

ЗАТВЕРДЖУЮ

Начальник кафедри інформаційних
технологій та телекомунікаційних
систем, к.т.н., доцент

Олександр ПРИДАТКО

«__» _____ 20__ р.

МЕТОДИЧНА РОЗРОБКА

для проведення практичного заняття з студентами (курсантами) 1-го курсу спеціальності 122 «Комп'ютерні науки» з дисципліни «Основи програмування»

Практичне заняття 5.6: «Написання лінійних програм. Застосування операторів»

Мета: здобути навички щодо роботи з основними типами даних та операторами в мові програмування Java.

Час: 2 годин

Місце заняття: комп'ютерна лабораторія

Матеріальне забезпечення:

- ПК з відповідним ПЗ та доступом до мережі Internet;
- Методичні рекомендації щодо виконання практичної роботи.

Після практичного заняття студенти (курсанти) повинні:

вміти: створювати елементарні програми із лінійним сценарієм виконання.

знати: основні типи даних та особливості застосування основних операторів.

Література:

1. Віртуальне навчальне середовище ЛДУ БЖД «Віртуальний університет» [Електронний ресурс]. – Доступний з : <http://virt.ldubgd.edu.ua/course/view.php?id=1778>
2. Програмування на Java (рос.) [Електронний ресурс]. – <http://kostin.ws/java>

План заняття:

1. Оголошення та ініціалізація змінних різних типів та застосування основних операторів
2. Видача індивідуальних практичних завдань

1. Оголошення та ініціалізація змінних різних типів

Для реалізації перших застосунків використовуючи основні типи даних розглянемо декілька розповсюджених прикладів.

Перед початком реалізації програми, в середовищі IDE Eclipse необхідно створити новий клас Arithmetic із автоматичною генерацією main методу (головного методу).

Приклад 1: Оголошення змінної типу `int`, ініціалізація її значенням та виведення у консоль:

```
public static void main(String[] arg) {  
    int counter;  
    counter = 99;  
    System.out.println(counter);  
}
```

Спростивши синтаксис зазначеного виразу можемо отримати:

```
public static void main(String[] arg) {  
    int counter = 99;  
    System.out.println(counter);  
}
```

Результат роботи:

99

Приклад 2: Застосування основних арифметичних операцій. В ході написання зазначеного прикладу рекомендовано почергово виводити у консоль результат виконання арифметичних операцій:

```
public static void main(String[] arg) {  
    int a;  
    a = 23 + 87;  
    System.out.println("Результат : "+a);  
    int b = a - 44;  
    System.out.println("Результат : "+b);  
    int c = a * b;  
    System.out.println("Результат : "+c);  
    double d = c / 53;  
    System.out.println("Результат : "+d);  
    int e = c / 53;  
    System.out.println("Результат : "+e);  
    int f = b % 10;  
    System.out.println("Результат : "+f);  
    double g = (b + a) / (a * 11.5);  
    System.out.println("Результат : "+g);  
}
```

Результат роботи:

Результат : 110

Результат : 66

Результат : 7260

Результат : 136.0

Результат : 136

Результат : 6

Результат : 0.1391304347826087

Приклад 3: Застосування операторів з присвоєнням та посилкового типу даних String:

```
public static void main(String[] arg) {  
    int x = 0;  
    String s = "Значення x рівне: ";  
    x += 20;  
    System.out.println(s+x);  
    x -= 15;  
    System.out.println(s+x);  
    x *= 2;  
    System.out.println(s+x);  
}
```

Результат роботи:

Значення x рівне: 20

Значення x рівне: 5

Значення x рівне: 10

Приклад 4: Застосування операторів інкремента та декремента. Спочатку постфіксні оператори (зверніть увагу на результат виконання програми):

```
public static void main(String[] arg) {  
    int x, a;  
    x = 10;  
    a = x++;  
    System.out.println("x=" + x);  
    System.out.println("a=" + a);  
    a = x--;  
    System.out.println("x=" + x);  
    System.out.println("a=" + a);  
}
```

Результат роботи:

x=11

a=10

x=10

a=11

Аналогічні операції з префісними операторами (зверніть увагу на результат виконання програми):

```
public static void main(String[] arg) {  
    int x, a;  
    x = 10;  
    a = ++x;  
    System.out.println("x=" + x);  
    System.out.println("a=" + a);  
    a = --x;  
    System.out.println("x=" + x);  
    System.out.println("a=" + a);  
}
```

```
}
```

Результат роботи:

```
x=11  
a=11  
x=10  
a=10
```

Ще один цікавий приклад з інкрементами та декрементами:

```
public static void main(String[] arg) {  
    int b = 5;  
    int a = 12;  
    int c = ++a - b++;  
    System.out.println(c);  
    int d = b++ * 2;  
    System.out.println(d);  
}
```

Результат роботи:

```
8  
12
```

Приклад 5: Особливості роботи з типом даних char:

```
public static void main(String[] arg) {  
    char a = 'Л';  
    char b = 'Д';  
    char c = 'У';  
    char d = 'Б';  
    char e = 'Ж';  
    char f = 'Д';  
    System.out.print(a);  
    System.out.print(b);  
    System.out.print(c);  
    System.out.print(d);  
    System.out.print(e);  
    System.out.print(f);  
}
```

Результат роботи:

```
ЛДУБЖД
```

Приклад 6: Застосування логічних типів даних та їх операторів:

```
public static void main(String[] arg) {  
    boolean a1 = true;  
    boolean a2 = false;  
    boolean a3 = a1 || a2;  
    System.out.println(a3);  
  
    boolean b1 = true;
```

```

    boolean b2 = false;
    boolean b3 = b1 && b2;
    System.out.println(b3);

    boolean c = true;
    boolean d = false;
    boolean e = true;
    boolean result = c && (d || e);
    System.out.println(result);

    boolean g = true;
    boolean h = !g;
    System.out.println(h);
}

```

Результат роботи:

```

true
false
true
false

```

Приклад 7: Застосування коментарів:

```

public static void main(String[] arg) {

    /*
    Приклад багаторядкового коментаря
    Застосування логічних операторів
    */

    boolean c = true; // Приклад однорядкового коментаря
    boolean d = false;
    boolean e = true;
    boolean result = c && (d || e);
    System.out.println(result); // виведення результату у консоль

}

```

2. Видача індивідуальних практичних завдань

2.1. Завдання:

Вкінці практичного заняття студенти (курсанти) отримують індивідуальне практичне завдання, результат виконання якого необхідно завантажити в віртуальне навчальне середовище. Індивідуальне практичне завдання полягає у створенні простих програмних додатків із використанням основних операторів мови Java.

Завдання 1. Створити програму з двома змінними типу `int` таким чином, щоб вона виводила у консоль результат їх ділення та залишку від ділення.

Завдання 2. Створити програму, яка виводитиме у консоль суму цифр двозначного числа записаного до змінної типу `int`.

Завдання 3. Створити програму, яка виводитиме у консоль округлене число, записане до змінної типу `double`, до найближчого цілого.

2.2. Здача індивідуальних завдань:

Практичні завдання завантажити у власний git-репозиторій. Для цього в папці проєкту в командній стрічці задайте команду **git init**. Після виконання цієї команди у папці проєкту має утворитись прихована папка **.git**.

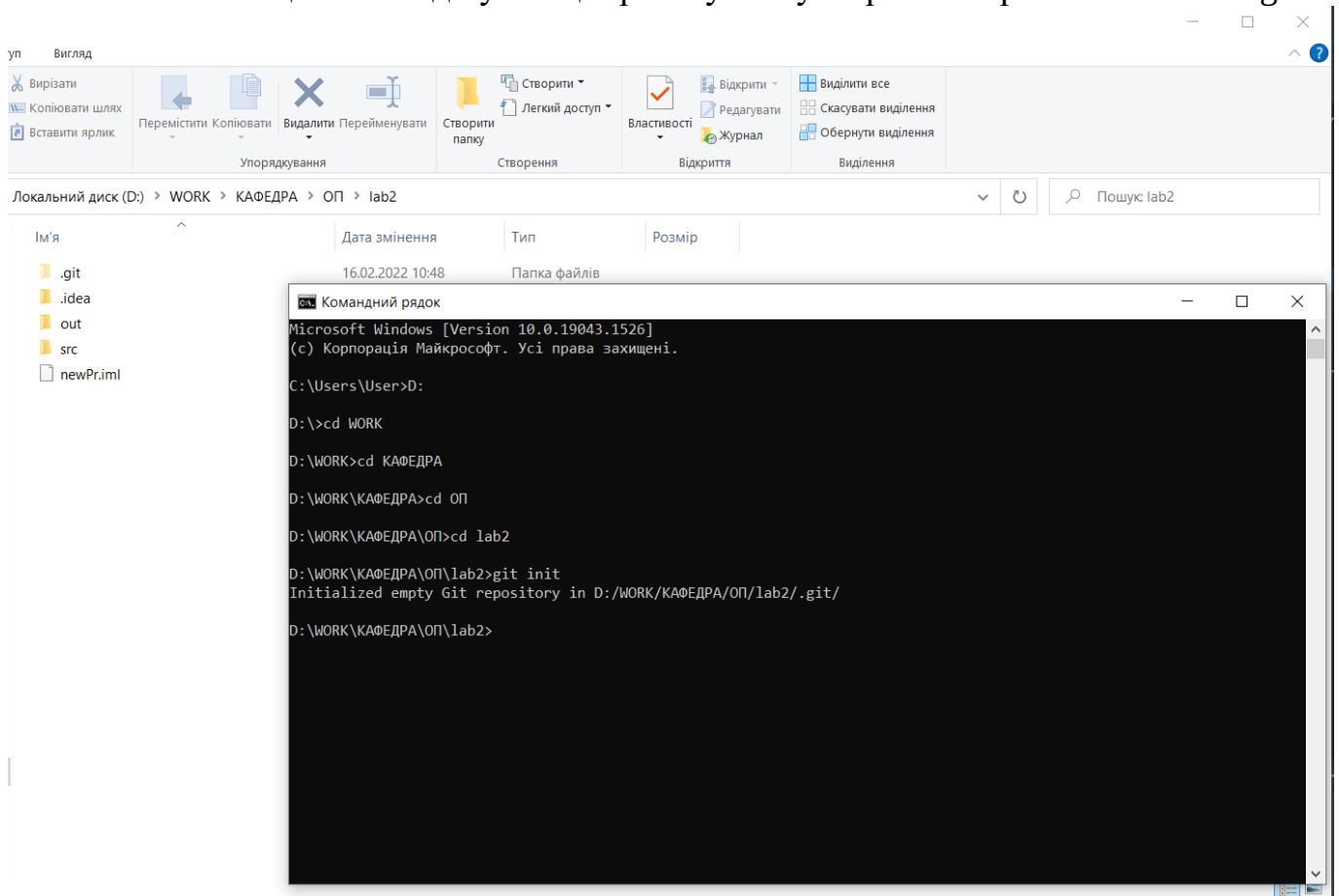


Рисунок 1 – Створення Локального репозиторію

Для зручності створимо файл **.gitignore**, в яку помістимо всі системні файли та папки середовища розробки.

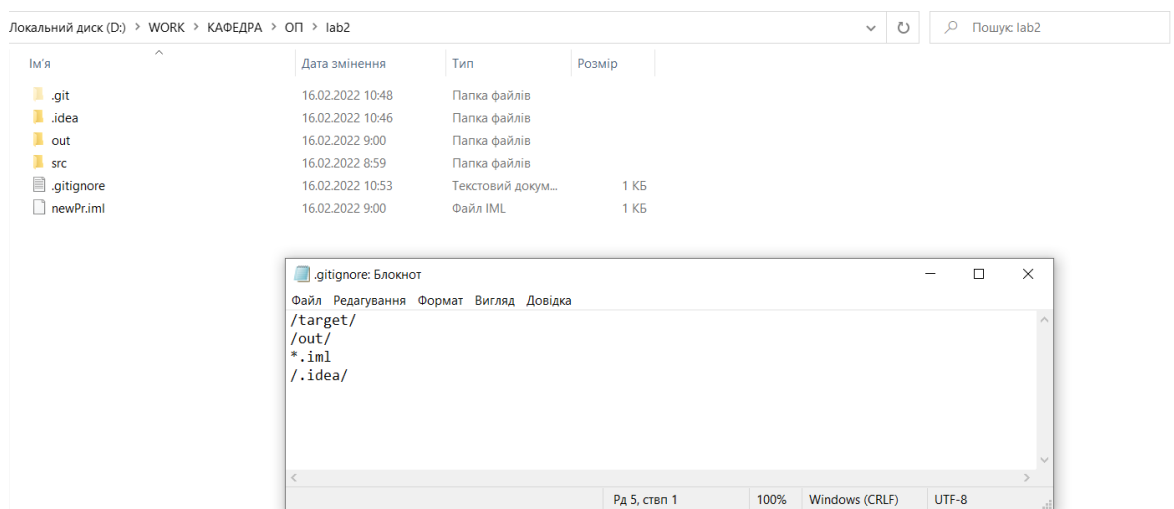


Рисунок 2 – Створення .gitignore

Після цього, потрібно створити глобальний репозиторій на GitHub та додати в нього локальний.

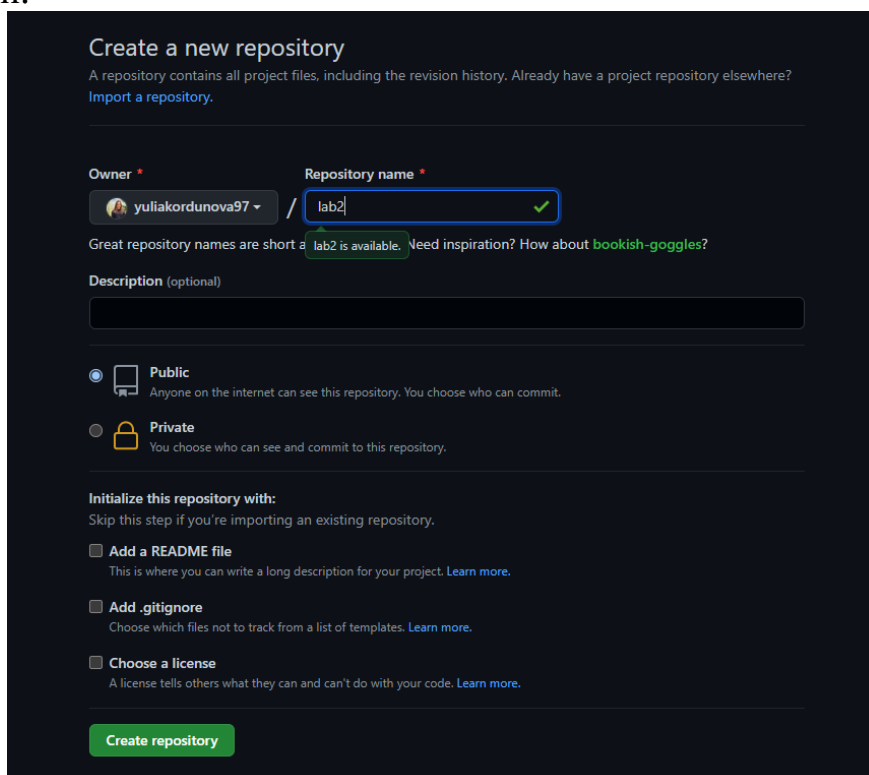


Рисунок 3 – Створення глобального репозиторію

Після цього в командному рядку виконуєте команди:

```
D:\WORK\КАФЕДРА\ОП\lab2>git remote add origin https://github.com/yuliakordunova97/lab2.git
D:\WORK\КАФЕДРА\ОП\lab2>git branch -M main
D:\WORK\КАФЕДРА\ОП\lab2>git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        src/

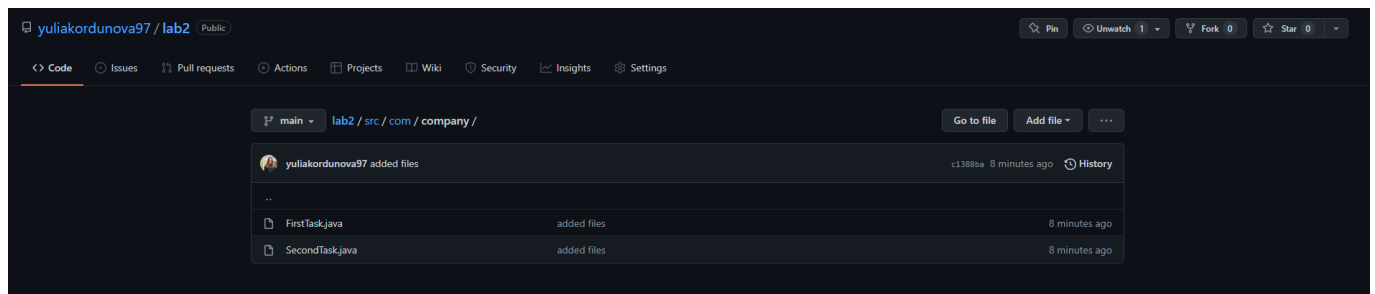
nothing added to commit but untracked files present (use "git add" to track)
```

```

D:\WORK\КАФЕДРА\ОП\lab2>git add .
D:\WORK\КАФЕДРА\ОП\lab2>git commit
[main (root-commit) c1388ba] added files
 3 files changed, 19 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 src/com/company/FirstTask.java
 create mode 100644 src/com/company/SecondTask.java
D:\WORK\КАФЕДРА\ОП\lab2>git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 656 bytes | 656.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yuliakordunova97/lab2.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
D:\WORK\КАФЕДРА\ОП\lab2>

```

Як результат отримаємо (оновивши сторінку на github):



Посилання на гіт репозиторій помістити в текстовий (.txt / .word) документ та завантажити у віртуальне середовище.

Методичну розробку розробив:
 Начальник кафедри ІТтаТС
 підполковник служби цивільного захисту

Олександр ПРИДАТКО

Методична розробка обговорена на засіданні кафедри ІТтаТС
 Протокол № ____ від " ____ " _____ 20__ р.