

ЗАНЯТИЕ 1.2

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ.

Тема: Создание простого Java-приложения

Задание: разработать консольную Java-программу по заданию в соответствии с вариантом.

Вариант 1. Разработать программу, вычисляющую площади поверхностей и периметры сторон для прямоугольника и треугольника. Размеры сторон вводит пользователь с клавиатуры.

Вариант 2. Разработать программу – банковский калькулятор вклада для вычисления ежемесячного дохода и итогового дохода. Начисление процентов происходит ежемесячно. Длительность вклада (в месяцах), **годовой** процент по вкладу и начальную сумму вводит пользователь с клавиатуры. При выполнении задания обязательно использовать цикл **for**.

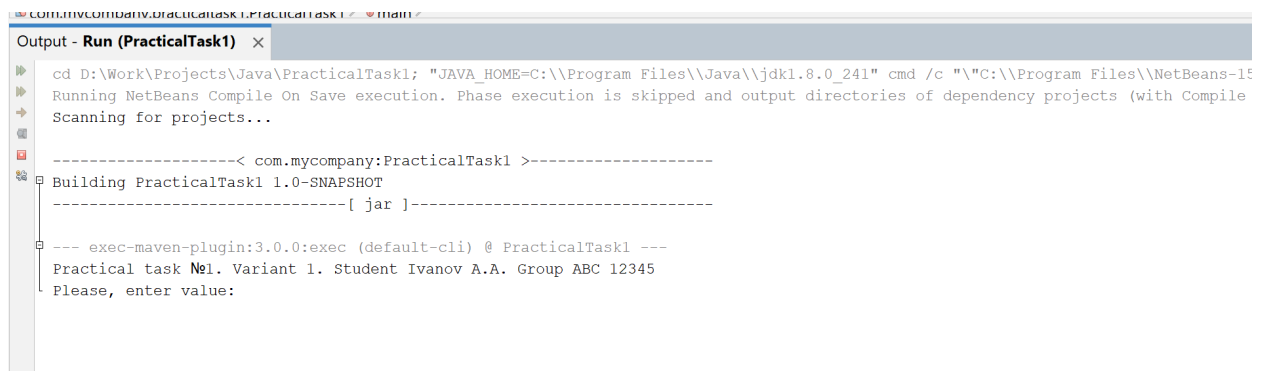
Вариант 3. Разработать программу по поиску четных и нечетных чисел в заданном диапазоне. Начальное и конечное значения диапазона должны быть **трехзначными**. Эти числа вводит пользователь с клавиатуры. Если пользователь ввел недопустимое значение числа, то необходимо отобразить сообщение об ошибке. При выполнении задания обязательно использовать цикл **for**.

Вариант 4. Разработать программу – банковский калькулятор вклада для вычисления ежемесячного дохода и итогового дохода. Начисление процентов происходит ежемесячно. Длительность вклада (в месяцах), **месячный** процент по вкладу и начальную сумму вводит пользователь с клавиатуры. При выполнении задания обязательно использовать цикл **while**.

Вариант 5. Разработать программу вычисления продолжительности работы в секундах в зависимости от дня недели. Рабочая неделя – пятидневная. С понедельника по четверг рабочий день – 8 часов, пятница – 6 часов. Суббота и воскресенье – выходные дни. Порядковый номер дня недели вводит пользователь с клавиатуры. Если пользователь ввел недопустимое значение (например, больше 7), то необходимо отобразить сообщение об ошибке. При выполнении задания обязательно использовать оператор **switch**.

Требования:

- 1) На сдачу практического задания отводится 7 дней.
- 2) Вариант определяется согласно порядковому номеру студента в журнале.
- 3) В случае дистанционного выполнения практического задания код программы необходимо опубликовать на Github и предоставить ссылку на него.
- 4) Итоговая программа должна компилироваться без ошибок, полностью выполнять требуемый функционал и при старте выводить в консоль: номер варианта, номер группы и ФИО студента (пример ниже):



```
cd D:\Work\Projects\Java\PracticalTask1; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_241" cmd /c "%C:\\Program Files\\NetBeans-15
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile
Scanning for projects...

-----< com.mycompany:PracticalTask1 >-----
Building PracticalTask1 1.0-SNAPSHOT
-----[ jar ]-----

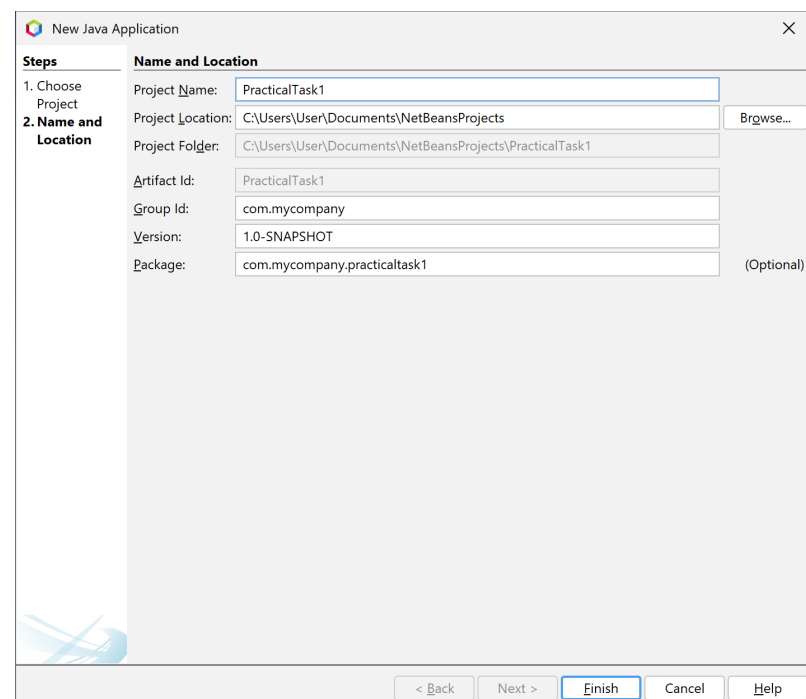
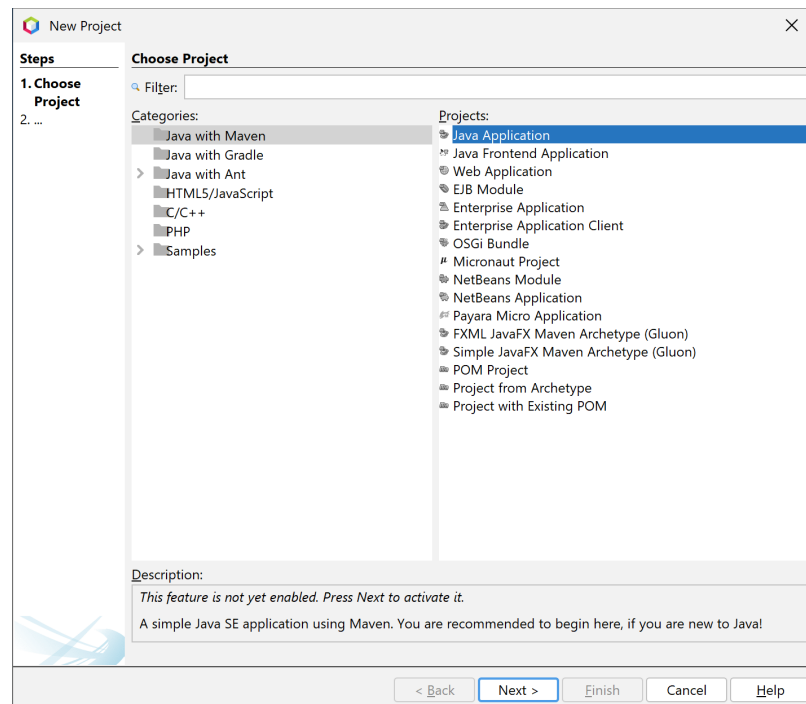
--- exec-maven-plugin:3.0.0:exec (default-cli) @ PracticalTask1 ---
Practical task №1. Variant 1. Student Ivanov A.A. Group ABC 12345
Please, enter value:
```

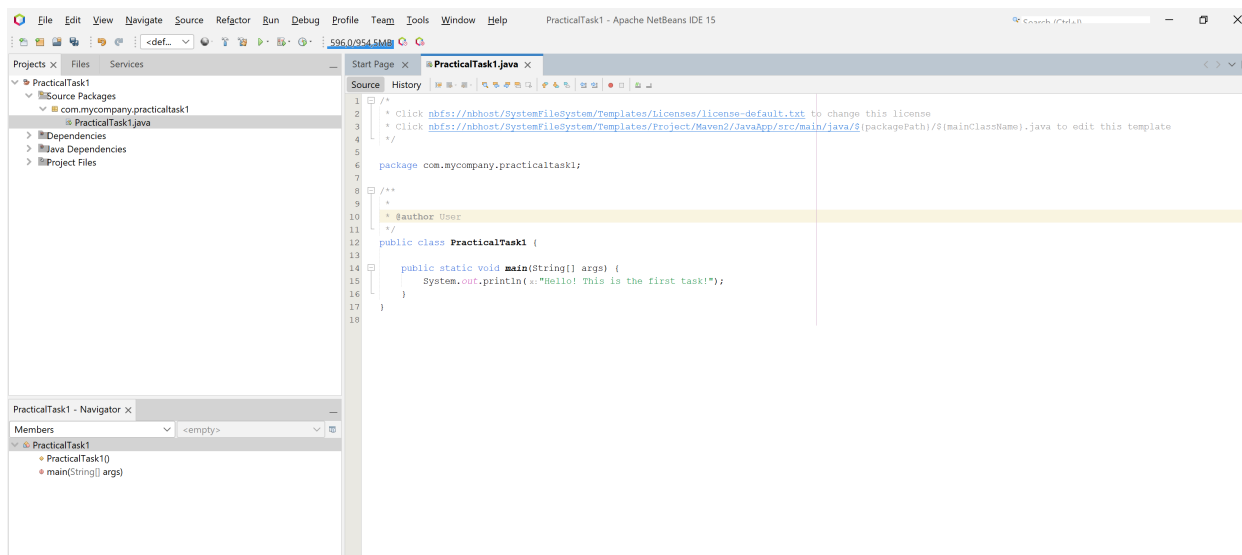
Дополнительная информация:

- 1) Перед выполнением задания рекомендуется ознакомиться с Лекциями №1.1.
- 2) Для считывания значений с клавиатуры необходимо использовать класс Scanner (java.util.Scanner).

```
System.out.println(x:"Please, enter int value:");
Scanner scan = new Scanner(in: System.in);
int value = scan.nextInt();
```

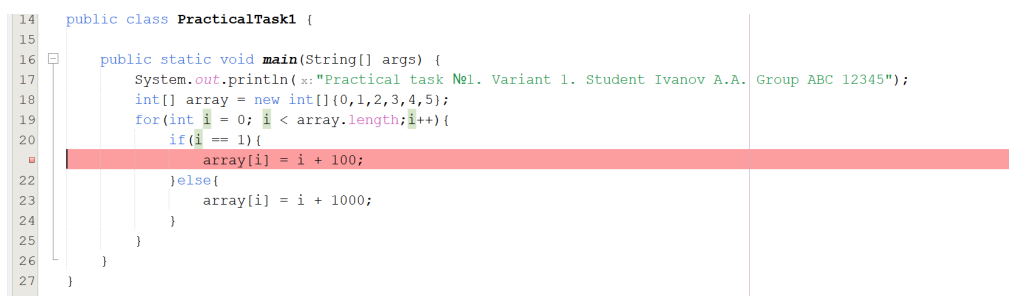
- 3) Для вычисления квадратного корня необходимо использовать метод Math.sqrt().
- 4) Порядок создания каркаса приложения в среде разработки NetBeans представлен на рисунках ниже.



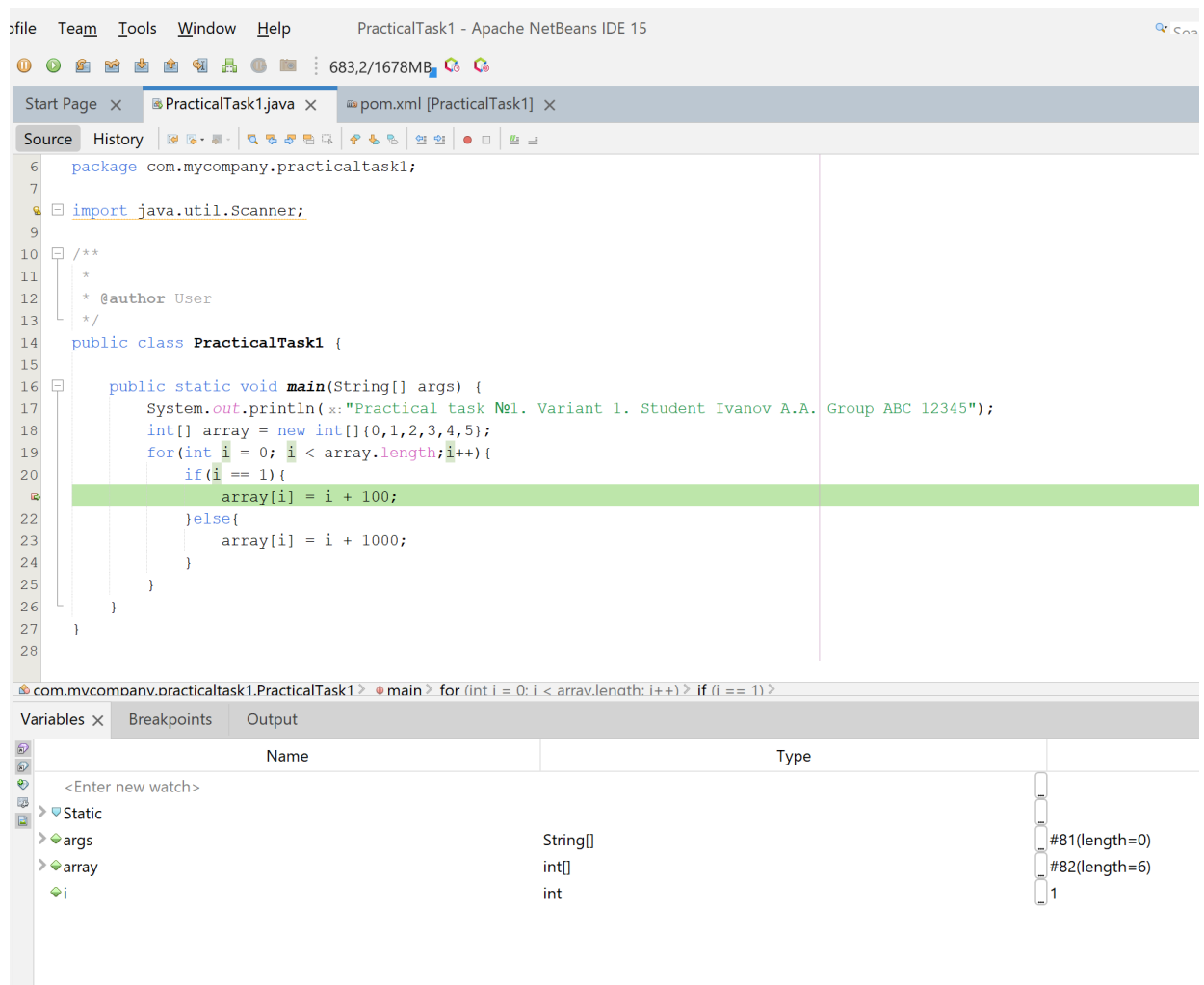


Для запуска разработанного приложения и проверки его работы необходимо нажать клавишу F6.

5) Отладка, или *debugging*, — это поиск (локализация), анализ и устранение ошибок в программном обеспечении в режиме реального времени (в ходе выполнения программы). Порядок отладки приложения по шагам в IDE Netbeans представлен на рисунках ниже.



Устанавливаем (левый щелчок мыши по полю слева) «breakpoint» на строке программы, где необходимо приостановить выполнение программы (строка выделяется красным цветом). Запуск приложения для отладки осуществляется с помощью комбинации клавиш Ctrl+F5. При достижении точки останова («breakpoint») строка подсвечивается зеленым цветом и в поле **Variables** доступны значения видимых на текущий момент переменных (при необходимости можно добавить для отслеживания другие переменные):



Для выполнения шага (одной операции) необходимо нажать клавишу F8, для продолжения работы программы до достижения следующей точки останова (если она есть, иначе – продолжить работу программы) необходимо нажать клавишу F5.