

Задания

<https://github.com/EreminD/course-27.12/blob/main/src/main/resources/task/conditions/L5.Conditions.md>

1. if

- Создайте переменную `balance` типа `int`.
- Положите в нее значение
- Если значение переменной равно 10, напишите "Десятка"

2. if-else

- Создайте переменную типа `int` и присвойте ей любое значение
- Если число четное (делится на 2 без остатка), напишите "Четное число."
- Если число нечетное, напишите "Нечетное число."

Чтобы проверить, что число четное, нужно выяснить, делится ли оно на 2 без остатка. Для этого существует оператор `%` – он возвращает остаток от деления. Например:

```
// в переменную запишется 1. Потому что 16:3 = 5 и один в остатке.  
int result = 16 % 3;
```

```
// в переменную запишется 0. Потому что 16:4 = 4 и ноль в остатке.  
int result = 16 % 4;
```

```
// в переменную запишется 0. Потому что 10:2 = 5 и ноль в остатке.  
int result = 10 % 2;
```

```
// в переменную запишется 1. Потому что 11:2 = 5 и один в остатке.  
int result = 11 % 2;
```

3. if-else + if

- Доработайте решение второго задания:
 - если число четное И делится на 4 без остатка, напишите "Четное число. Кратно четырем"
 - если число нечетное И делится на 3 без остатка, напишите "Нечетное число. Кратно трем."

4. Проверка пароля

- Переменная `String` с именем `password`
- Опишите алгоритм проверки пароля. Требования к паролю:

- длина пароля не меньше 8 символов. Иначе написать ошибку "Пароль не менее 8 символов"
- пароль должен содержать минимум одну цифру. Иначе написать ошибку "Пароль должен содержать минимум 1 цифру"
- пароль должен содержать минимум 1 из символов [!, @, #, \$, %, ^, &, *, №]. Иначе написать ошибку "Пароль должен содержать минимум 1 спецсимвол"
- если пароль подходит по всем критериям, написать "пароль принят"

5. Задание со звездочкой. FizzBuzz

Это – классическое задание с собеседования.

- выведите на экран числа 1 до 100. При этом:
 - если число кратно 3, напечатайте Fizz
 - если число кратно 5, напечатайте Buzz
 - если число кратно и 3, и 5, напечатайте FizzBuzz

Результат будет выглядеть так:

```
1
2
Fizz
4
Buzz
...
13
14
FizzBuzz
16
...
98
Fizz
Buzz
```

<https://github.com/EreminD/course-27.12/blob/main/src/main/resources/task/oop/L5.Methods.md>

1. Маскировка карты 2.0

- Создайте класс Card
- У класса должны быть поля:
 - номер
 - срок годности
 - cvv
 - pinCode
- Поля класса должны задаваться через конструктор
- Опишите в классе метод, который печатает номер карты в формате "**** * 3456"
- Опишите в классе метод, который принимает на вход пин-код. Если переданный пин-код совпал с пин-кодом карты, напечатать в консоль номер карты без маски (все цифры)
- Создайте класс MyProgram + метод psvm
 - В методе создайте новую карту
 - Вызовите сначала метод печати с маской
 - Потом метод печати с пин-кодом
 - Убедитесь, что нельзя получить никакие данные карты напрямую через поля (используйте private)

2. Разгон автомобиля

- Создайте класс Car
- Поля могут быть любые.
- Обязательно должно быть поле currentSpeed (тип данных int). По умолчанию currentSpeed = 0;
- Создайте метод getCurrentSpeed. Метод должен возвращать (return) текущую скорость
- Создайте метод speedUp, который принимает на вход число. Метод должен увеличивать текущую скорость на заданное число.
- Создайте метод break(), который сокращает текущую скорость на 10 единиц.
 - Если текущая скорость ≤ 10 , то метод делает текущую скорость равной 0
- Создайте класс MyProgram + метод psvm.
- В методе создайте новый экземпляр класса.
- Запросите текущую скорость и выведите ее на экран.
- Увеличьте скорость на 25 единиц.
- Запросите текущую скорость и выведите ее на экран. Должно быть 25
- Вызовите метод brake()
- Запросите текущую скорость и выведите ее на экран. Должно быть 15
- Вызовите метод brake()
- Запросите текущую скорость и выведите ее на экран. Должно быть 5

- Вызовите метод brake()
- Запросите текущую скорость и выведите ее на экран. Должно быть 0

Пример кода работы с классом Car:

```
Car car = new Car("Volvo", "S60");

String speed = car.getCurrentSpeed();
System.out.println(speed); // тут 0

car.speedUp(25);
speed = car.getCurrentSpeed();
System.out.println(speed); // тут 25

car.brake()
speed = car.getCurrentSpeed();
System.out.println(speed); // тут 15

car.brake()
speed = car.getCurrentSpeed();
System.out.println(speed); // тут 5

car.brake()
speed = car.getCurrentSpeed();
System.out.println(speed); // тут 0
```