

# Zadanie rekrutacyjne

*Napisz program w języku Java zgodny z następującymi wymaganiami*

- Program operuje na figurach geometrycznych, umożliwiając wyliczenie pola figury na podstawie danych przekazanych do algorytmu.
- Komunikacja z programem odbywa się poprzez konsolę, jednakże w przyszłości może to zostać zastąpione przez inny interfejs (np. formularz WWW).

*Interakcja z programem*

- `area [figura] [dane wejściowe]`, czyli `area square 2` wyświetli pole dla kwadratu o boku 2.
- Analogicznie `area rectangle 2 4` spowoduje obliczenie pola prostokąta o bokach 2 i 4.
- Na obecną chwilę program ma umożliwiać obliczenie pola kwadratu, prostokąta oraz koła, jednakże w przyszłości może to zostać rozbudowane o kolejne figury.
- Wynik każdej operacji ma zostać zapisany w "pamięci podręcznej programu" (na czas trwania sesji z użytkownikiem, kolejne uruchomienie programu powoduje wymazanie pamięci). Zawartość pamięci podręcznej może zostać wyświetlona za pomocą komendy `showall`.
- Komenda `showall` wypisuje zawartość bufora na ekran w formie:

Na obecną chwilę zaimplementowany ma być jeden domyślny raport, jednak w niedalekiej przyszłości dodane będą kolejne raporty: `dotted`, `dashed` (które od domyślnego raportu różnić się będą nagłówkiem zawierającym nazwę raportu oraz separatorami pól: kropki i kreski zamiast spacji). Wybór odpowiedniego raportu ma odbywać się poprzez dodatkowy parametr `showall dotted`. Komenda `showall` bez dodatkowych parametrów będzie wyświetlała domyślny raport opisany w poprzednim punkcie.

*Poza komendami `area` oraz `showall` aplikacja powinna umożliwiać*

Wyświetlenie powitania **Welcome to Area Calculator. Please enter a command.**

- Powitanie powinno wyświetlać się w momencie uruchomienia programu. Powitanie zawsze można powtórzyć poprzez wydanie komendy `hello`

Zakończenie pracy poprzez wydanie komendy `exit`

W przyszłości system może zostać potencjalnie rozbudowany o kolejne komendy, np. `saveall` oraz procedurę liczenia obwodu figury geometrycznej. Zaplanuje architekturę rozwiązania w taki sposób aby spełnienie tych wymagań nie wymagało nadmiernych nakładów pracy.

Aby ułatwić zadanie przygotowany został podstawowy bazowy projekt implementujący komunikację poprzez konsolę z wykorzystaniem klasy `Scanner`. Docelowe rozwiązanie może, lecz nie musi z niego skorzystać

```

package com.example.test;

import java.util.Scanner;

public class AreaCalculator {

    public static void main(String[] args) {

        boolean ok = true;
        Scanner s = new Scanner(System.in);

        while (ok) {
            System.out.print("calculator > ");
            String commandString = s.nextLine();

            System.out.println("Command: " + commandString);
        }
        s.close();
    }
}

```

#### TIP

Pamiętaj, że najważniejsze jest pokazanie zrozumienia problemu oraz umiejętnego ujęcia go w odpowiednie praktyki programowania. Szczególną uwagę zwróć na architekturę rozwiązania, która ma umożliwić możliwe najłatwiejszą implementację wymienionych przyszłych wymagań. Jeżeli obawiasz się, że nie starczy Tobie czasu nie skupiaj się na implementacji poszczególnych algorytmów a na ogólnej budowie aplikacji i prawidłowym jej przetestowaniu (także automatycznym).