ZMIENNE

Pytania

1. Jak należy zdefiniować zmienną?

Zmienne definiujemy podając, kolejno, ich typ, nazwę, oraz ewentualną wartość początkową. Możemy zdefiniować więcej, niż jedną zmienną na raz, oddzielając ich nazwy przecinkami:

```
int pierwszaCyfra = 8;
char a;
double x, y;
```

2. Czy zmiennej trzeba nadać wartość początkową w momencie definicji?

Nie trzeba, ale przed użyciem takiej zmiennej należy jej przypisać wartość. Jeżeli spróbujemy użyć zmiennej zdefiniowanej w metodzie zanim nadamy tej zmiennej wartość, to nasz program w ogóle się nie skompiluje:

```
public class UzycieNiezainicjalizowanejZmiennej {
        public static void main(String[] args) {
                int x;
                // blad! nie nadalismy zmiennej x jeszcze zadnej wartosci
                System.out.println("Wartosc x wynosi: " + x);
        }
}
```

3. Do czego służą operatory +=, -=

Są to skrótowe operatory przypisania. Dla przykładu, operator += powoduje dodanie do zmiennej po lewej stronie operatora wartości wyrażenia po jego prawej stronie:

```
Int a = 2;
a += 3; // a bedzie mialo wartosc 2 + 3, czyli 5
```

4. Do czego służą operatory ++ i -- oraz czym się różnią ich post- i pre-fixowe wersje?

   Operatory te służą do zwiększania oraz zmniejszania wartości zmiennej o 1. Post-fixowe wersje tych operatorów różnią się tym od pre-fixowych, że najpierw wracają wartość zmienianej zmiennej, a dopiero potem ją zmieniają. Pre-fixowe wersje najpierw zmieniają wartość zmiennej, a potem zwracają jej wartość:

```
int x = 2;
int y = 2;
System.out.println(x++); // wyswietli 2
System.out.println(++y); // wyswietli 3
```

5. Jaką wartość będzie miała zmienna x, a jaką zmienna y, w poniższym przykładzie?

```
int x = 5++;
int y = ++5;
```

   Kod się nie skompiluje, operator ++ oczekuje zmiennej jako argumentu, a nie literału liczbowego.

6. Co zostanie wyświetlone na ekranie w wyniku wykonania poniższego programu?

```
String dzien = "Poniedzialek!";
powitanie.toUpperCase();
System.out.println(dzien);
```

Na ekranie wyświetlony zostanie komunikat "Poniedzialek!". Użyliśmy na zmiennej ***dzien*** metody, która zamienia znaki w stringu z małych na duże, ale nie przypisaliśmy wyniku działania tej metody do żadnej zmiennej.
Metoda toUpperCase nie modyfikuje oryginalnej zmiennej, lecz zwraca nową wartość z małymi literami zamienionymi na wielkie.

7. Co zostanie wypisane w wyniku działania poniższego programu?

```
public class WyswietlX {
        public static void main(String[] args) {
                int x;
```

```
                System.out.println("Wartosc x " + x);
        }
}
```

Kod w ogóle się nie skompiluje, ponieważ nie zainicjalizowaliśmy zmiennej x żadną wartością. Kompilator jest to w stanie wykryć i zgłosi błąd już na etapie kompilacji.

8. Jaką wartość będzie miała zmienna `wynik`?

```
int wynik = 2.5 * 20;
```

Kod w ogóle się nie skompiluje. Wynik działania 2.5 * 20 to liczba rzeczywista (ponieważ jeden z argumentów operatora * to liczba rzeczywista), a zmienna, do której ten wynik próbujemy przypisać, to zmienna typu **int**. Zmienne typu **int** mogą przechowywać jedynie liczby całkowite.

9. Co zostanie wypisane w wyniku działania poniższego programu?

```
public class WyswietlXY {
        public static void main(String[] args) {
                int x = 10;
                int y = -5;
                System.out.println("Wspolrzedne X i Y to: " + X + ", " + Y);
        }
}
```

Kod się nie skompiluje – kompilator zgłosi błąd, że nie wie, czym są X oraz Y. Zmienne, które wcześniej zdefiniowaliśmy, zapisaliśmy małymi literami – w języku Java wielkość znaków ma znaczenia.

10. Czy poniższy kod skompiluje się poprawnie?

```
char a = "A";
System.out.println(a);
```

Nie, ponieważ do zmiennej typu **char** próbujemy przypisać łańcuchów znaków. Chociaż złożony tylko z jednego znaku, nadal jest to łańcuch znaków, a nie pojedynczy znak. Aby kod był poprawny, do zmiennej powinien zostać przypisany znak ujęty w apostrofy 'A'.

Zadania

1. Dodawanie

Napisz program, w którym zdefiniujesz trzy zmienne typu `int`*. Do dwóch pierwszych przypisz dowolne liczby, a do trzeciej – wynik dodawania dwóch pierwszych liczb. Aby dodać do siebie wartości dwóch zmiennych, skorzystaj ze znaku +. Wyświetl wynik na ekranie.*

2. Obwód trójkąta

Napisz program, który skorzysta z czterech zmiennych w celu policzenia obwodu trójkąta. W trzech zmiennych zapisz długość każdego z boków, a do ostatniej zmiennej przypisz wynik – obwód trójkąta. Wyświetl wynik na ekranie.

3. Aktualna data

Napisz program, w którym do trzech różnych zmiennych przypiszesz aktualny dzień, miesiąc, i rok. Pamiętaj o odpowiednim nazewnictwie zmiennych. Wypisz na ekran wszystkie wartości.

4. Inicjały

Napisz program, w którym przypiszesz swoje inicjały do dwóch zmiennych typu `char` *– do każdej ze zmiennych po jednym znaku. Wypisz swoje inicjały na ekran – po każdej literze powinna następować kropka, np. A. P.*

5. Obwód trójkąta

Napisz program, który pobierze od użytkownika trzy boki trójkąta, policzy jego obwód i wypisze wynik na ekran.

Aby pobrać od użytkownika liczby całkowite, należy dodać do kodu programu instrukcję `import` oraz metodę `getInt`.

6. Słowa w odwrotnej kolejności

Napisz program, który wczyta od użytkownika trzy słowa i wypisze je w odwrotnej kolejności, niż podał je użytkownik, oddzielone przecinkami.

Jeśli użytkownik poda

1. *jeden*

2. *dwa*
3. *trzy*

to program powinien wyświetlić trzy, dwa, jeden

*Tym razem musimy skorzystać z metody **getString**, dzięki której będziemy mogli pobierać od użytkownika słowa (musimy także skorzystać z instrukcji import).*

7. Wielkie litery

*Napisz program, który pobierze od użytkownika słowo i wypisze je z małymi literami zamienionymi na wielkie. Skorzystaj z metody **toUpperCase** typu String.*

*Ponownie skorzystaj z metody **getString** w celu pobrania słowa od użytkownika.*