

Zadania

Zadanie 1

(Konstruktor)

Co wydrukuje poniższy kod?

```
class ABC {  
  
    public static void main(String[] args) {  
        Wiosna w = new Wiosna();  
    }  
}  
  
class Wiosna {  
    wiosna() {  
        System.out.println("Witaj wiosno!");  
    }  
}  
}
```

Zadanie 2

(kontruktor, this)

Utwórz konstruktor w klasie *Szkolenie*, tak by instrukcje w metodzie *main* mogły wykonać się prawidłowo.

```
class ABC {  
  
    public static void main(String[] args) {  
        Szkolenie sz = new Szkolenie(20);  
        System.out.println(sz.liczbaUczestnikow);  
    }  
}  
  
class Szkolenie {  
    int liczbaUczestnikow;  
    KOD_TUTAJ  
}
```

Zadanie 3*

(kompozycja)

Zaimplementuj klasę *Szampon*, tak by instrukcje na obiekcie *Kapiel* k mogły się prawidłowo wykonać.

```
public class ABC {
    public static void main(String[] args) {
        Kapiel k = new Kapiel();
        k.ileMamPlynu();
        k.dolejPlyn(60);
        k.ileMamPlynu();
        k.dolejPlyn(140);
        k.ileMamPlynu();
        k.ileMamSzamponu();
        k.dolejSzamponu(20);
        k.ileMamSzamponu();
        k.dolejSzamponu(30);
        k.ileMamSzamponu();
    }
}

class Kapiel {
    PlyndoKapieli pkp;
    Szampon sz;
    Kapiel() {
        pkp = new PlyndoKapieli();
        sz = new Szampon();
    }
    public void dolejPlyn(int ile) {
        System.out.println("Dolano do kapieli: " + pkp.wylejIlosc(ile));
    }
    public void ileMamPlynu() {
        System.out.println("Stan plynu do kapieli: " + pkp.sprawdzIloscPlynu());
    }
    public void dolejSzamponu(int ile) {
        sz.wylej(ile);
    }
    public void ileMamSzamponu() {
        System.out.println(sz.sprawdzIloscSzamponu());
    }
}

class PlyndoKapieli {
    private int ilosc;
    PlyndoKapieli() {
        this.ilosc = 100;
    }
    public int wylejIlosc(int ilosc) {
        if (this.ilosc == 0) { // nie mam z czego wylac
            return 0;
        }
        int ileDoWydania = 0;
        if (ilosc >= this.ilosc) {
            ileDoWydania = this.ilosc;
            this.ilosc = 0;
        } else if (ilosc < this.ilosc) {
            ileDoWydania = ilosc;
        }
    }
}
```

```

        this.ilosc -= ilosc;
    }
    return ileDoWydania;
}
public int sprawdzIloscPlynu() {
    return ilosc;
}
}

class Szampon {
    private int ilosc;
    Szampon() {
        this.ilosc = 100;
    }
    KOD_TUTAJ
}

```

Zadanie 4

(dziedziczenie)

Zaimplementuj klasy Samochod, Motorowka oraz Motor, tak by kod metodzie main wydrukował wartości zamieszczone w komentarzu.

```

public class ABC {
    public static void main(String[] args) {
        Samochod sam = new Samochod();
        Motorowka motorowka = new Motorowka();
        Motor motor = new Motor();

        System.out.println(sam.sprawdzKolor()); // czerwony
        System.out.println(motorowka.sprawdzKolor()); // czerwony
        System.out.println(motor.sprawdzKolor()); // czerwony

        sam.sprawdzIloscKol(); // 4
        motorowka.sprawdzIloscKol(); // 0
        motor.sprawdzIloscKol(); // 2
    }
}

class Pojazd {
    public void sprawdzIloscKol() {
    }
    public String sprawdzKolor() {
        return "czerwony";
    }
}

class Samochod {

```

```
}  
  
class Motorowka {  
  
}  
  
class Motor {  
  
}
```

Zadanie 5

(projektowanie interfejsu)

Spróbuj stworzyć deklaracje metod dla obsługi plików.

```
interface Plik {  
  
}
```

Zadanie 6

Dopisz kod, który wskaże, że klasa *ObslugaPlikow* implementuje interfejs *Plik*.

```
interface Plik {  
    public void utworzPlik(String sciezkaINazwaDoPliku);  
    public void wczytajPlik(String sciezkaINazwaDoPliku);  
    public void zapiszPlik(String sciezkaINazwaDoPliku, String daneDoZapisania);  
    public void usunPlik(String sciezkaINazwaDoPliku);  
}  
  
class ObslugaPlikow {  
  
    public void utworzPlik(String sciezkaINazwaDoPliku) {  
        System.out.println("Utworzyłem plik: " + sciezkaINazwaDoPliku);  
    }  
  
    public void wczytajPlik(String sciezkaINazwaDoPliku) {  
        System.out.println("Wczytałem plik: " + sciezkaINazwaDoPliku);  
    }  
  
    public void zapiszPlik(String sciezkaINazwaDoPliku, String daneDoZapisania) {  
        System.out.println("Zapisałem do pliku: " + sciezkaINazwaDoPliku);  
    }  
  
    public void usunPlik(String sciezkaINazwaDoPliku) {  
        System.out.println("Usunalem plik: " + sciezkaINazwaDoPliku);  
    }  
}
```

```
}
```

Zadanie 7

Uruchomienie poniższego kodu spowoduje wyrzucenie wyjątku.

Odczytaj rodzaj wyjątku.

Napisz kod, który obsłuży wyjątek dotyczący tablic a następnie zapewni obsługę wyjątków dowolnych.

```
public static void main(String args[]) {  
    int arr[] = {1,2,3,4,5};  
    System.out.println(arr[7]);  
}
```

Zadanie 8

Uruchom poniższy kod i sprawdź czy rozwiązanie z zadania 7 poprawnie obsłuży wyjątek dzielenia przez zero. Tym razem indeks w tablicy nie zgłosi błędu.

```
int arr[] = {1,2,3,4,5};  
System.out.println(arr[2]);  
System.out.println(4/0);
```

Zadanie 9

Dopisz kod, który wymusi na programiście obsługę wyjątków w metodzie main.

Użyj menu kontekstowego STS do otoczenia wywołania metody otworzPlik(); blokiem try catch.

```
public static void main(String args[]) {  
    otworzPlik();  
}  
  
public static void otworzPlik() {  
    String pathname = "C:\\Users\\Wlodek\\Desktop\\abc.txt";  
    File file = new File(pathname);  
    FileReader fileReader = new FileReader(file);  
    BufferedReader reader = new BufferedReader(fileReader);  
    String line = null;  
    while ((line = reader.readLine()) != null) {  
        System.out.println(line);  
    }  
}
```