

# Program do dozowania pigmentów do farb

Do sprawnie działającego programu, program powinien zawierać:

Listę dostępnych pigmentów:

Ilość 12 pigmentów, które program przechowuje w bazie danych.  
Każdy pigment ma swój osobny zbiornik, z którego program będzie pobierał odpowiednią ilość w zależności od wybranego przez użytkownika koloru.

Listę dostępnych farb (lista tylko i wyłącznie może być utworzona przez „producenta”, zwykły użytkownik nie może mieć dostępu do jej zmiany):

- Farby poliuretanowe – ich nazwę, pojemność, kod bazy (czy transparentna – do ciemnych kolorów, czy biała – do jasnych kolorów)
  - Farby epoksydowe – jak wyżej
- Impregnaty – występuje tylko i wyłącznie jeden kod bazy

Listę dostępnych kolorów:

- Kolory, które jako kod bazy farby mają (TCH, TCL, TVH, TAL) – przeznaczone do pigmentacji farb:
  - poliuretanowych
  - epoksydowych

W zależności od wybrania farby program powinien przeliczać stosunek dozowania pigmentów do wybranej farby, np.:

Do farby epoksydowej, program powinien przeliczyć 50% pigmentów, które są dozowane do farb poliuretanowych.

Kolory, które jako kod bazy farby mają (TCW) – przeznaczone do pigmentacji farb:

- dla impregnatów

Funkcje programu:

- Użytkownik wybiera farbę z listy.
- Użytkownik wpisuje kolor, który chce zamieszać.
- Program powinien pokazać ilość pigmentów do dozowania, oraz kod bazy farby.
  - Użytkownik wpisuje ilość farb do zakolorowania.
- Program wyświetla informacją jaką nazwę farby oraz kod bazy podstawić pod mieszalnik.
  - Użytkownik zatwierdza.

- Następuje proces dozowania.
- Po zakończeniu dozowania program odejmuje od ilości pigmentu w zbiorniku, pigmenty, które zostały zadozowane do farby.
  - Użytkownik w każdej chwili może anulować bieżące zadanie.
- Użytkownik może sam również stworzyć swój własny kolor, podając pigment, rodzaj farby, oraz kod bazy.

Klasa główna tworzenie obiektu:

```
class Paints:
    def __init__(self, id, name, capacity, value, first_base, second_base, type_of_paint):
        self.id = id
        self.name = name
        self.capacity = capacity
        self.value = value
        self.first_base = first_base
        self.second_base = second_base
        self.type_of_paint = type_of_paint
```

Podklasa – tworzenie obiektu - farby poliuretanowe:

```
from paints_paints_class import Paints

class PolyurethanePaints(Paints):
    def __init__(self, id, name, capacity, value, first_base, second_base, type_of_paint):
        super().__init__(id, name, capacity, value, first_base, second_base, type_of_paint)
```

Podklasa – tworzenie obiektu - farby epoksydowe:

```
from paints_paints_class import Paints
import paints_mysql

class EpoxyPaints(Paints):
    def __init__(self, id, name, capacity, value, first_base, second_base, type_of_paint):
        super().__init__(id, name, capacity, value, first_base, second_base, type_of_paint)
```

Podklasa – tworzenie obiektu – impregnaty:

```
from paints_paints_class import Paints
```

```
class ImpregnatingPaints(Paints):
```

```
    def __init__(self, id, name, capacity, value, first_base, second_base, type_of_paint):
```

```
        super().__init__(id, name, capacity, value, first_base, second_base, type_of_paint)
```