

Utworzyć klasę `RowerName`, zawierającą nazwę roweru `char * name`. Dla umieszczenia wiersza tekstowego z nazwą roweru wykonać dynamiczne alokowanie pamięci o wymaganej liczbie symboli.

Utworzyć klasę `RowerPrice`, zawierającą cenę roweru `double price`.

Utworzyć klasę `Rower`: `class Rower : public RowerName, public RowerPrice`. Klasa `Rower` nie posiada swoich własnych zmiennych.

Utworzyć funkcje-szablon

`template <class T> void PrintToScreen(const T& ob, const char* title)`, gdzie `ob` – obiekt dowolnego typu, `title` – wiersz tekstowy. Funkcja ta wyprowadzi na monitor obiekt `ob`.

Przystosować klasy `RowerName`, `RowerPrice`, `Rower` do poprawnego działania w kodzie:

```
int main()
{
    vector<Rower> vect;
    Rower tab_Rowery[] = { Rower("Kross", 1500), Rower("Rover", 2000), Rower("Giant", 2100), Rower("Unibike", 1800), Rower("Kross",
1500) };
    const size_t no_elem = sizeof(tab_Rowery) / sizeof(Rower);

    //Put elements of tab_Rowery in vect

    //output elements of vect to screen
    PrintToScreen(vect, "before sorting");

    //Find all occurrences of Rower objects with the name "Kross" using algorithm std::find.

    //sort elements of vector according names of bikes in descending order

    //output elements of vect to screen
    PrintToScreen(vect, "after sorting");

    //remove all elements from vector vect with the duplicated bike names.
```

```
    //output elements of vect to screen
    PrintToScreen(vect, "finally:");

    vect.clear();
}
```

Po zakończeniu proszę przesłać archiwum projektu przez czat MS TEAMS.