

Zadanie 2 - obiektowe wzorce projektowe (2pkt)

W grze dostarczamy moduł do rozgrywki, w której biorą udział czołgi jednego typu. Czołgi T72, Twardy posługują się amunicją Missile125, zaś czołgi Leopard amunicją Missile120.

- Dodaj fabrykę aby tworzyć zgodne zestawy (czołg/amunicja).
- Dodaj możliwość tworzenia zestawu dla T72 (nie jest w hierarchii Tank), nie można modyfikować klasy T72. Składowa id przechowuje dokładnie to samo, co name w klasach z hierarchii Tank.

```
class Tank {
public:
    Tank(const string& name) : name_(name) {}
    virtual ~Tank() {}
    const string& getName() const { return name_; }
private:
    string name_;
};
class LeopardTank : public Tank {
public:
    LeopardTank(const string& name) : Tank(name) {}
};
class TwardyTank : public Tank {
public:
    TwardyTank(const string& name) : Tank(name) {}
};

class Missile {
public:
    Missile();
    virtual ~Missile() {}
};
class Missile125 : public Missile { };
class Missile120 : public Missile { };

class T72 {
    T72(const string& id) : id_(id) {}
    const string& getId() const { return id_; }
private:
    string id_;
};
```

```
using UTank = unique_ptr<Tank>;
using UMissile = unique_ptr<Missile>;
class AbstractFactory {
public:
    virtual UTank createTank(const std::string& name) = 0;
    virtual UMissile createMissile() = 0;
};
```

Uwagi do prowadzącego (R. Nowaka):

Zadanie 5 - UML (2pkt)

Proszę odpowiedzieć na następujące pytania:

- Czy to jest dobre wymaganie (jeśli nie – to dlaczego): *System powinien wspierać śledzenie wymagań pomiędzy SSS a SRS oraz SRS a SDD.*

- Po co definiuje się metody weryfikacji wymagań? O jakie metody warto zabiegać i dlaczego?

Uwagi do prowadzącego (K. Grochowskiego):