DARP dokumentace

# Úvod

Aplikace DARP je naprogramována v jazyce C# ve frameworku .NET 6 pro operační systém Microsoft Windows. Vzhledem k tomu, že .NET 6 funguje i na dalších platformách, jako UNIX či MacOS, lze aplikaci do bucnou rozšířit i na ně. Protože je logika aplikace striktně oddělena od UI, bude takové rozšíření snadné.

Hlavní projektem je solution je DARP. Zdrojové soubory v něm jsou rozděleny do složek – logických celků. Níže popíšeme jednotlivé logické celky a souvislosti mezi nimi a v DARP docs.xml je pak k dispozici detailní dokumentace jednotlivých typů a metod generovaná z kódu.

# Models

Zde jsou všechny důležité datové struktury, které se používají napříč aplikací. Některé jsou technické povahy, jako *Cords, Time* či *TimeWindow*. Další, jako *Vehicle* či *Order* slouží pouze pro jako reprezentace daného konceptu a neobsahují interní logiku. Naopak třídy *Route* a *Plan* implementují několik obecných metod, které jsou pro úlohu plánování společné. Přesněji, *Plan* je seznam objektů typu *Route* a implementuje jednak metodu pro výpočet celkového profitu a jednak pro aktualizaci pozic vozidel a jejich tras při posunu v čase. *Route* pak implementuje metody pro výpočet profitu trasy a pro ověření možnosti vložení a vložení objednávky do trasy.

# Services

Zde jsou tzv. služby, které slouží pro obsluhu dat. Přístup k službám zprostředkovává *ServiceProvider.* Každá služba je implementací obecného kontraktu a takových implementací může existovat více. Například můžeme mít implementaci pro lokální uložení dat pro testování a zároveň implementaci téže služby pro ukládání dat do databázového uložiště.

# Solvers

Zde najdeme implementace konkrétních algoritmů pro řešení *Dial A Ride Problem.* Všechny implementují kontrakt *ISolver* a aplikaci tak lze snadno doplnit o další algoritmy. Implementace jednoho algoritmu se sestává vždy z minimálně tří tříd implementující kontrakty *ISolver*, *ISolverInput* a *ISolverOutput*. Jako výchozí implementace pro vstup a výstup lze využít třídy *SolverInputBase* a *SolverOutputBase*.

# Utils

Zde jsou různé podpůrné třídy pro ostatní části programu. Třída *LoggerBase* implementuje kontrakt *ILogger* a slouží k logování v programu. Logger zapisuje zprávy do kolekce TextWriteru. Jednou z implementací TextWriteru je *TextBoxWriter*, který vypisuje zprávy do komponenty *TextBox*. *LoggerBase* i *TextBoxWriter* jsou *thread safe*. Třída *XMath* pak implementuje další metody, které nejsou standartní knihovně *Math*.

# Windows

Zde je kód hlavního okna aplikace s logikou uživatelského rozhraní. Aplikace používá návrhový vzor MVVM.