



OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

СКАЧАНО С WWW.SW.HELP - ПРИСОЕДИНЯЙСЯ!

Проверить, идет ли запись!





Меня хорошо видно && слышно?

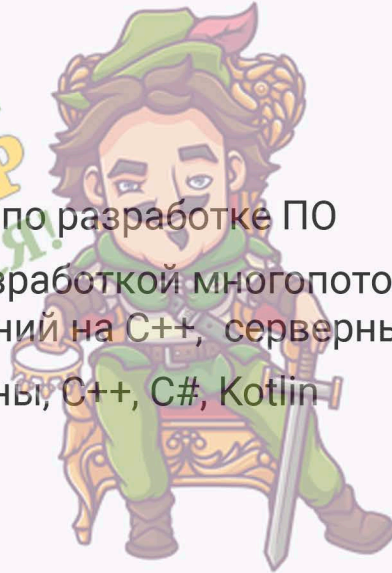
Ставьте , если все хорошо
Напишите в чат, если есть проблемы

Преподаватель



Тюменцев Евгений

- 9 лет руковожу компаний по разработке ПО
- в прошлом занимался разработкой многопоточных кросс-платформенных приложений на C++, серверных приложений на C#
- 20 преподаю ООП, паттерны, C++, C#, Kotlin



Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack #канал группы или #general

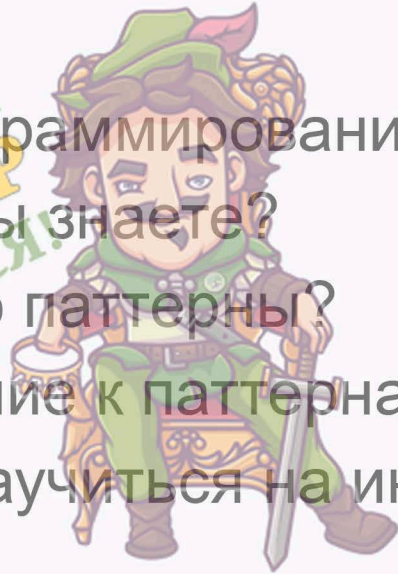


Вопросы вижу в чате, могу ответить не сразу

Знакомство

Напишите в чат:

1. Ваш опыт программирования
2. Какие языки Вы знаете?
3. Что знаете про паттерны?
4. Какое отношение к паттернам?
5. Чему хотите научиться на интенсиве?



Тайминг: 5 мин

Цели курса | После занятия вы сможете

1

определять места в программном коде, которые могут потенциально приводить к снижению скорости разработки ПО и образованию технического долга.

2

научиться выявлять и применять паттерны

Смысл | Зачем вам это уметь

1

Умение видеть проблемные места в коде – это возможность научиться ими управлять.



SOLID



Тюменцев Евгений

Генеральный директор

HWdTech LLC

etyumentcev@gmail.com

Маршрут вебинара

Решаем модельную задачу



SOLID



Улучшаем решение



Рефлексия

Цели вебинара | После занятия вы сможете

1

строить абстракции, удовлетворяющие SOLID

Смысл | Зачем вам это уметь

1

Чтобы писать код, устойчивый к изменениям требований

The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network pattern of dots and lines is overlaid across the center. The text is centered within this band.

**Кто знаком с SOLID
принципами?**

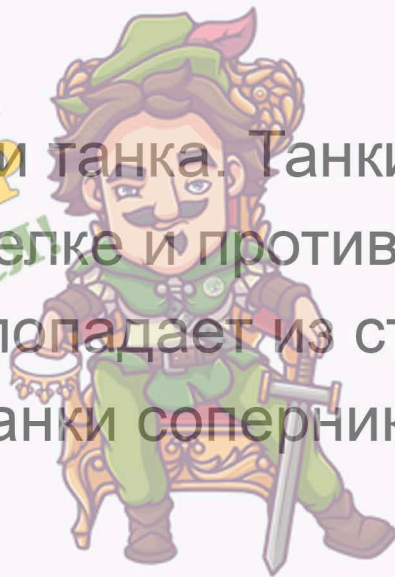


**Опишите Ваш опыт применения
SOLID?**



Игра Танки

С каждой стороны играют по три танка. Танки могут двигаться вперед, назад, поворачиваться по часовой стрелке и против часовой стрелке, стрелять. Танк выходит из строя, когда в него попадает из строя. Выигрывает та команда, которая выведет из строя все танки соперника.



Задание

Реализовать движение танка

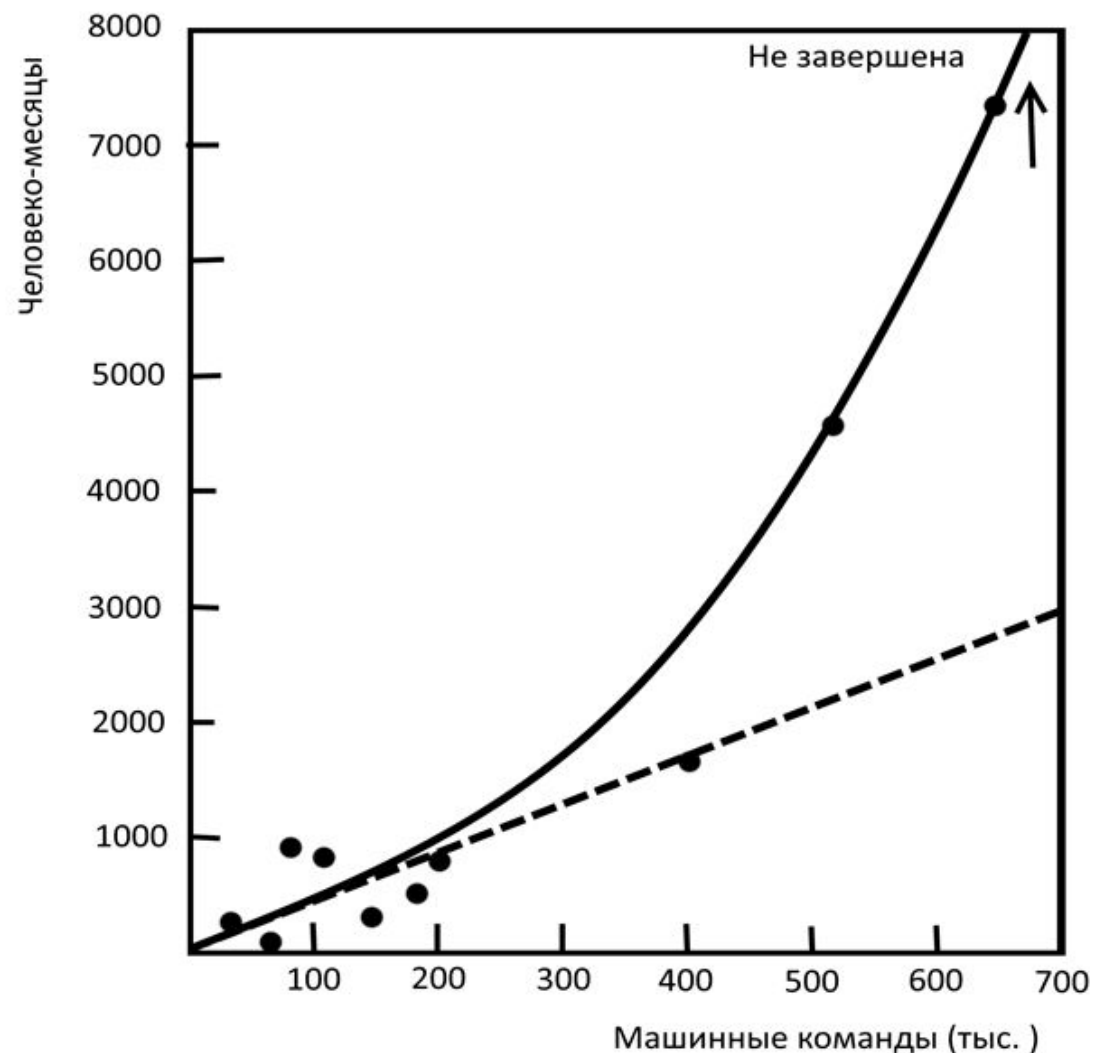


Тайминг: 20 мин

Обсуждаем решения

Если теперь придется реализовать движение снаряда рассмотрим + и - реализаций

Скорость разработки падает с ростом проекта



Nanus B., Farr L.

Some cost contributors to large-scale programs //

AFIPS Proc. SJCC. Spring 1964.
Vol. 25. P. 239-248.

Степенная функция с показателем 1.5

Проблема сложности разработки ПО

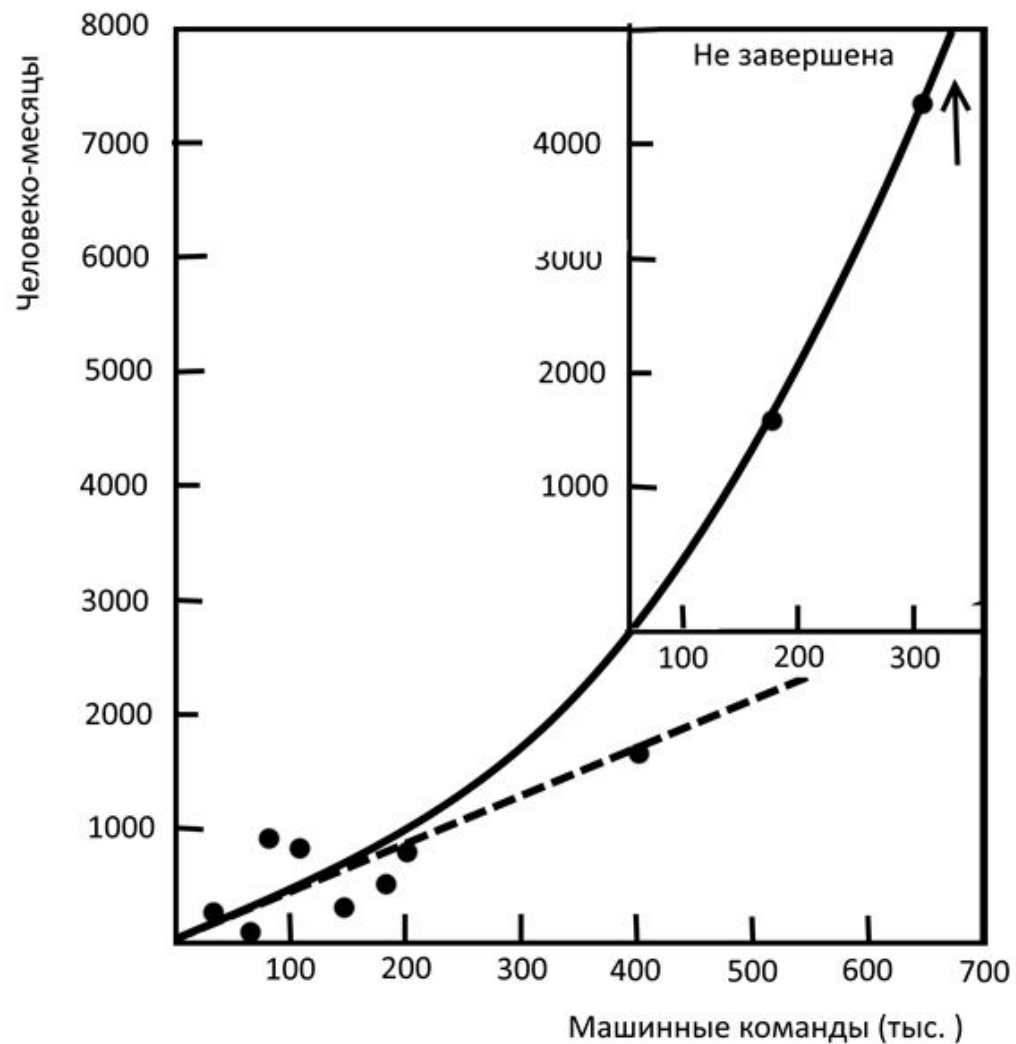
Потребность в ресурсах растет быстрее, чем размер задачи

«Нужно бежать со всех ног,
чтобы только оставаться на
месте, а чтобы куда-то попасть,
надо бежать как минимум
вдвое быстрее!»

- Алиса в стране чудес -



Берем готовые решения



Пытаемся запустить свою кривую как можно выше на старте



Agile



The Open-Closed Principle

Тюменцев Е.А. О формализации процесса разработки программного кода

<http://links.hwdtech.com/formal-definition-of-software-development-process>

The Open-Closed Principle

Программные объекты должны быть открыты для расширения, но в тоже время закрыты для модификации.

И что с эти делать?

!Это достаточное условие постоянной скорости разработки, в части «закреты для модификации»

!Справедливо для всех языков программирования

Напишите в чат

А может быть есть конструкции языков программирования, определенный код, которые могут провоцировать нарушение ОСР?

Проблему могут возникать с

copy-paste

switch

enum

if-else-if

явное приведение типа

new T

магические константы

Об абстракциях

1 Абстрагирование - это отображение множества сущностей на множество абстракций

2 Либо у нас не рассмотрены все кейсы, либо у нас используется обобщение

3 Либо обобщаем сущности до одной абстракции, либо будут правки



Ссылка

Тюменцев Е.А. "О формальном определении абстракции"// Математические структуры и моделирование 2018. No 1(45). С. 131–143.

Определение абстракции

Пусть Σ – произвольный алфавит, $Abs \subseteq \Sigma^*$ – некоторое подмножество Σ^* , X – произвольное множество.

Тогда сюръективное отображение $F: X \rightarrow Abs$ называется **абстрагированием** множества X над алфавитом Σ .

Элементы множества X – **сущности**.

Элементы множества Abs – **абстракции**.

Причина изменения абстракций

Пусть Σ – произвольный алфавит символов, $F : X \rightarrow A$, где $A \subset \Sigma^*$, – некоторое абстрагирование с обобщением множества X над алфавитом Σ , причём $|A| > 1$. Предположим, что Y – множество такое, что $|Y| > 1$, $id : Y \rightarrow AY$ – взаимно-однозначное абстрагирование множества Y над алфавитом Σ . Тогда существует такое отображение $G : X \rightarrow Y$, для которого не существует $g : A \rightarrow AY$, чтобы следующая диаграмма была коммутативной:

$$\begin{array}{ccc} X & \xrightarrow{F} & A \\ G \downarrow & & \downarrow g \\ Y & \xrightarrow{id} & AY \end{array}$$

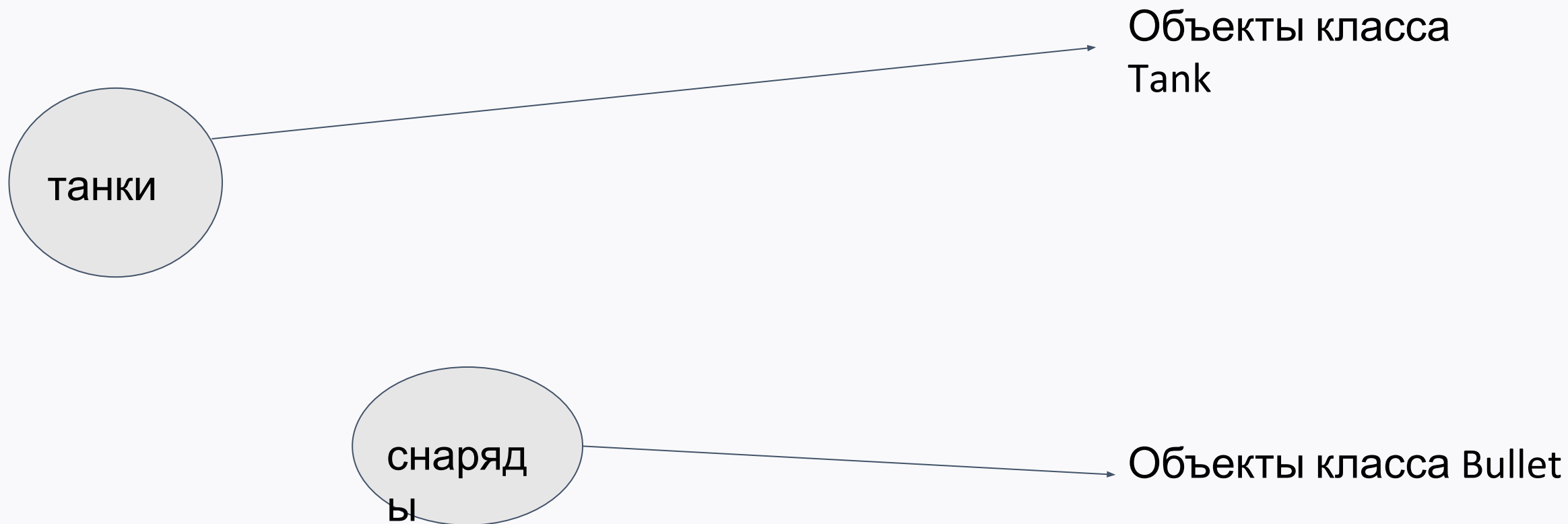
The Single-Responsibility Principle

Должна быть ровно одна причина для изменения класса.

The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network diagram pattern (nodes and connecting lines) is overlaid across the center. The text "Как выполнить SRP?" is written in white, bold, sans-serif font within this band.

Как выполнить SRP?

Абстрагирование движения



Об обобщении

Пусть Σ – произвольный алфавит символов, w – некоторое слово в этом алфавите, $F : X \rightarrow A$, где $A \subset \Sigma^*$, – некоторое абстрагирование множества X над алфавитом Σ .

Предположим, что слово w содержит все абстракции множества A в качестве подслов.

Тогда:

либо F является абстрагированием с обобщением,

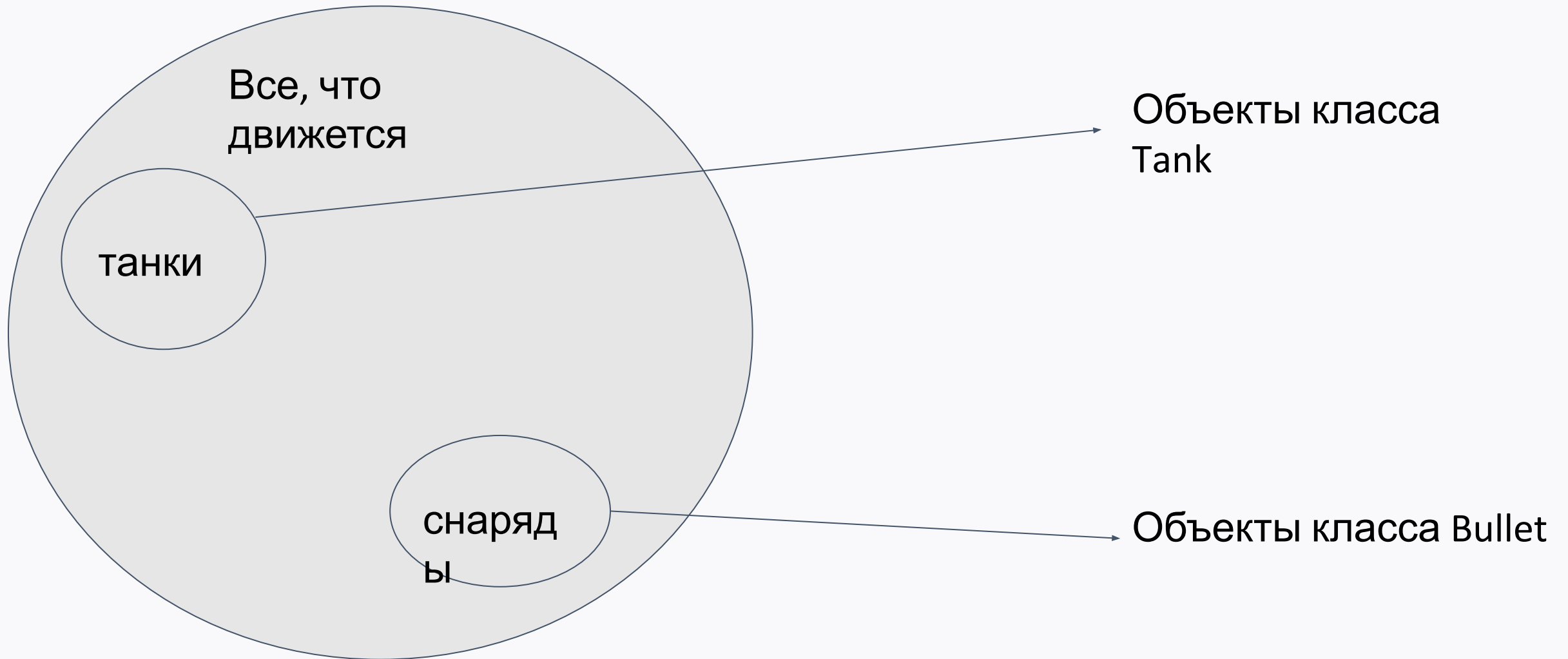
либо X – конечное множество,

либо F определено частично на множестве X .



У нас обобщение или частично определенное абстрагирование?

Абстрагирование движения



Причина изменения абстракций

Пусть Σ – произвольный алфавит символов, $F : X \rightarrow A$, где $A \subset \Sigma^*$, – некоторое абстрагирование с обобщением множества X над алфавитом Σ , причём $|A| > 1$. Предположим, что Y – множество такое, что $|Y| > 1$, $id : Y \rightarrow AY$ – взаимно-однозначное абстрагирование множества Y над алфавитом Σ . Тогда существует такое отображение $G : X \rightarrow Y$, для которого не существует $g : A \rightarrow AY$, чтобы следующая диаграмма была коммутативной:

$$\begin{array}{ccc} X & \xrightarrow{F} & A \\ G \downarrow & & \downarrow g \\ Y & \xrightarrow{id} & AY \end{array}$$

Абстрагирование движения

Все, что
движется


?

Задание

Реализовать движение танка. Версия 2



Тайминг: 20 мин

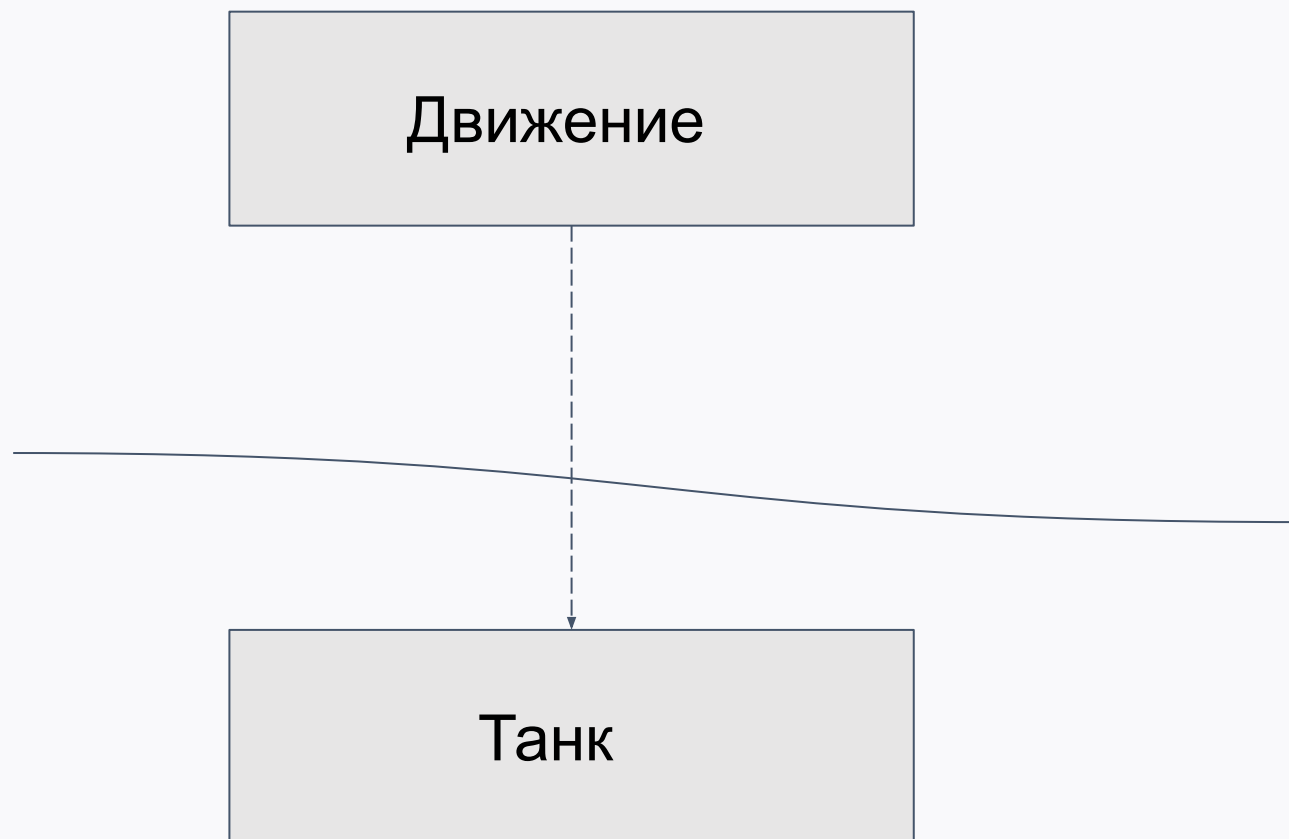
The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network pattern of dots and lines is overlaid across the center. The text is centered within this band.

Напишите +/- второй реализации?

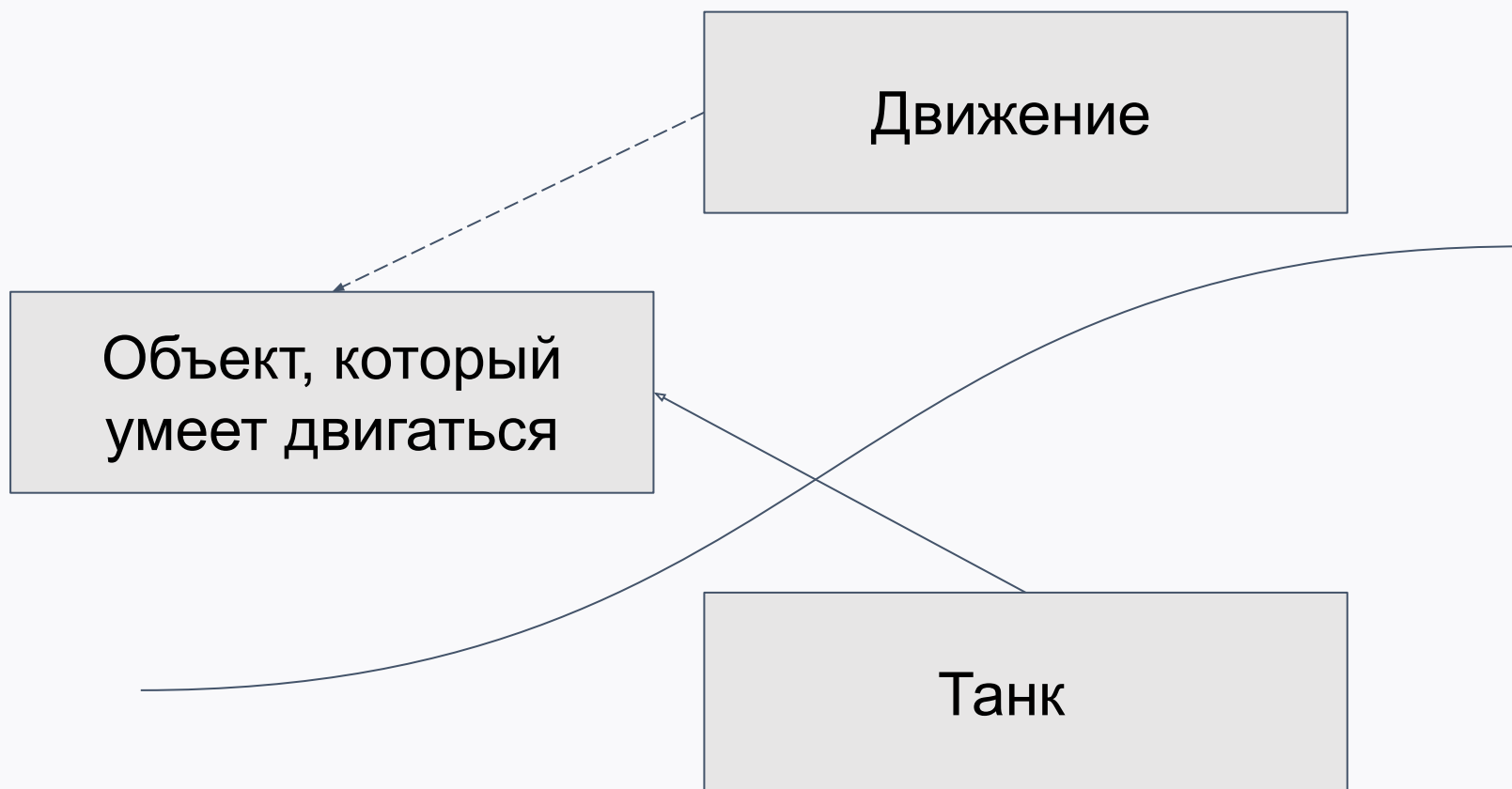
The Dependency Inversion Principle

*Высокоуровневые компоненты не должны зависеть от низкоуровневых компонент. И те, и те должны зависеть от абстракций.
Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.*

Первое решение



Второе решение



The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network diagram pattern (nodes and connecting lines) is overlaid across the middle of the image. The text is centered within this band.

**Как понять, что мы выбрали
правильные абстракции?**

The Liskov Substitution Principle

*Функции, которые **используют** ссылки на базовые классы, должны иметь возможность использовать объекты производных классов, не зная об этом*

The Interface Segregation Principle

Класс не должен зависеть от интерфейсов, которые он не использует.

Ответьте на вопросы

1 Зачем нужны SOLID принципы?

2 Если SOLID знали до вебинара, то поменялось ли Ваше отношение к SOLID принципам?

3 Что нужно сделать с множеством сущностей, чтобы код удовлетворял DIP?

Одна мысль на
лайде

без картинок

Использование
цифр для опроса

0-9



Маршрут вебинара

Решаем модельную задачу



Команда



Наша реализация



Рефлексия



Паттерн Команда



Тюменцев Евгений

Генеральный директор

HWdTech LLC

etyumentcev@gmail.com

Цели вебинара | После занятия вы сможете

- 1 применять паттерн Команда к задачам


СМЫСЛ | Зачем вам это уметь

1

Использовать готовые конструкции в фреймворках

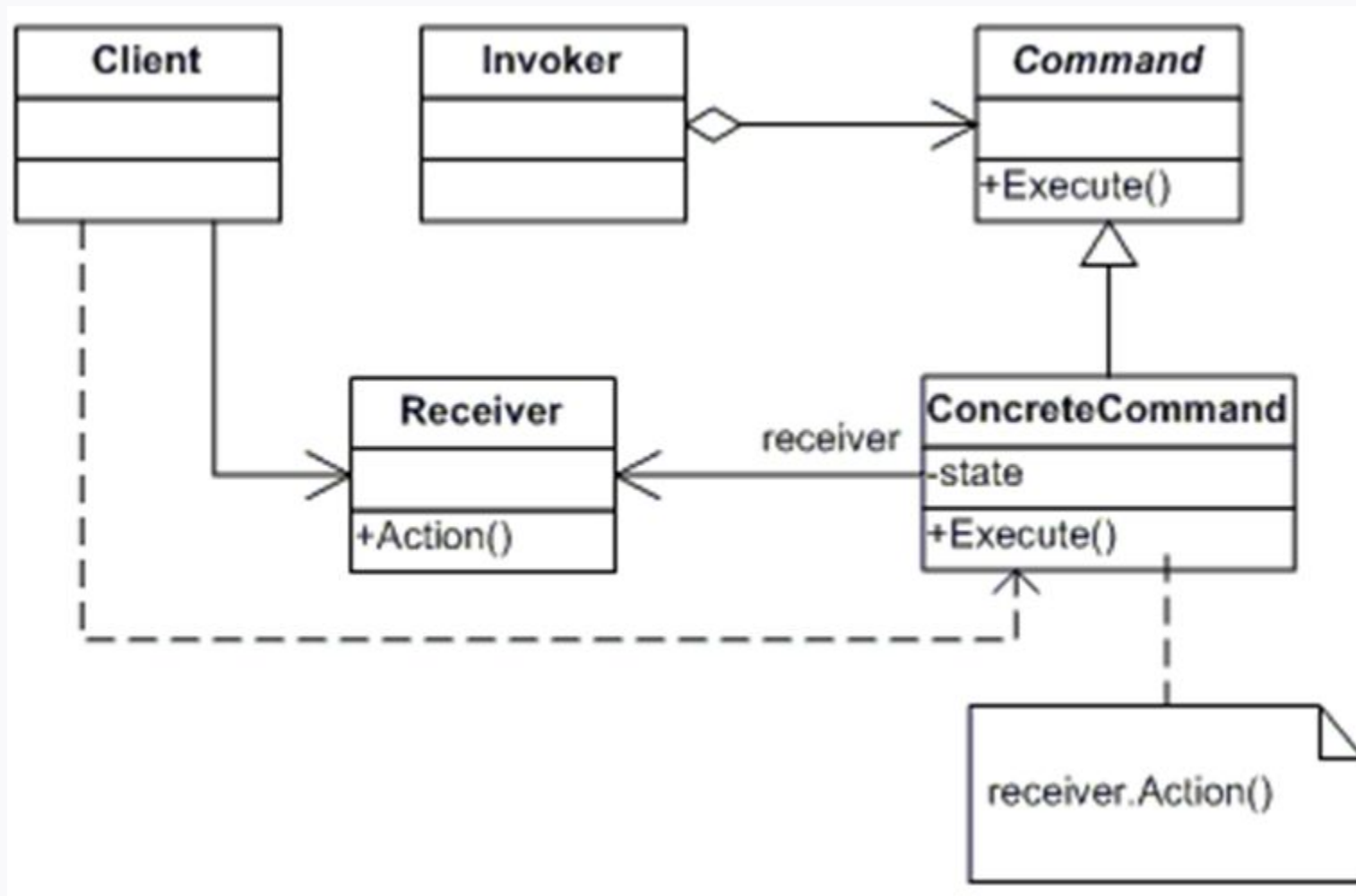
2

Улучшить повторную используемость кода

The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network pattern of dots and lines runs horizontally across the middle of the image. The text is centered within this band.

**Кто знаком с паттерном
команда?**

Диаграмма классов паттерна Команда



Сфера применения

UI

Сфера применения

UI

Запись макросов

Сфера применения

UI

Запись макросов

Undo/Redo

Сфера применения

UI

Запись макросов

Undo/Redo

Пулы потоков

Сфера применения

UI

Запись макросов

Undo/Redo

Пулы потоков

Транзакции

История развития паттерна

1. callback
2. Классы
3. Библиотечные generic-типы
4. Лямбда-функции

Callback

```
/* qsort example */
#include <stdio.h>    /* printf */
#include <stdlib.h>   /* qsort */

int values[] = { 40, 10, 100, 90, 20, 25 };

int compare (const void * a, const void * b)
{
    return ( *(int*)a - *(int*)b );
}

int main ()
{
    int n;
    qsort (values, 6, sizeof(int), compare);
    for (n=0; n<6; n++)
        printf ("%d ",values[n]);
    return 0;
}
```

```
void qsort (void* base, size_t num, size_t size,
            int (*compar)(const void*,const void*));
```

Классы

```
class Command
{
    public:
        virtual ~Command() {}
        virtual void execute() = 0;
    protected:
        Command( Game* p ): pgame( p) {}
        Game * pgame;
};
```

Библиотечные generic-типы

docs.microsoft.com/en-us/dotnet/api/system.func-7?view=netcore-3.1

Version

.NET Core 3.1

Search

- > [HttpStyleUriParser](#)
- [Func<TResult>](#)
- [Func<T,TResult>](#)
- [Func<T1,T2,TResult>](#)
- [Func<T1,T2,T3,TResult>](#)
- [Func<T1,T2,T3,T4,TResult>](#)
- [Func<T1,T2,T3,T4,T5,TResult>](#)
- [Func<T1,T2,T3,T4,T5,T6,TResult>](#)**
- [Func<T1,T2,T3,T4,T5,T6,T7,TResult>](#)
- [Func<T1,T2,T3,T4,T5,T6,T7,T8,TResult>](#)
- [Func<T1,T2,T3,T4,T5,T6,T7,T8,T9,TResult>](#)

specified by the `TResult` parameter.

In this article

- [Definition](#)
- [Remarks](#)
- [Extension Methods](#)
- [Applies to](#)
- [See also](#)

C# Copy

```
public delegate TResult Func<in T1,in T2,in T3,in T4,in T5,in T6,out TResult>(T1 arg1, T2 arg2, T3 arg3, T4 arg4, T5 arg5, T6 arg6);
```

Type Parameters

T1

The type of the first parameter of the method that this delegate encapsulates. This type parameter is contravariant. That is, you can use either the type you specified or any type that is less derived. For more information about covariance

Лямбда-функции

```
Func<int, int> square = x => x * x;  
Console.WriteLine(square(5));
```

Покритикуем реализацию

```
cmd->execute(5, "bla-bla-bla");
```

Что потенциально может поменяться в этой строке?

Покритикуем реализацию

```
cmd->execute(5, "bla-bla-bla");
```

Утверждение

Пусть Σ – произвольный алфавит символов, $F : X \rightarrow A$, где $A \subset \Sigma^*$, – некоторое абстрагирование с обобщением множества X над алфавитом Σ , причём $|A| > 1$. Предположим, что Y – множество такое, что $|Y| > 1$, $id : Y \rightarrow AY$ – взаимно-однозначное абстрагирование множества Y над алфавитом Σ . Тогда существует такое отображение $G : X \rightarrow Y$, для которого не существует $g : A \rightarrow AY$, чтобы следующая диаграмма была коммутативной:

$$\begin{array}{ccc} X & \xrightarrow{F} & A \\ G \downarrow & & \downarrow g \\ Y & \xrightarrow{id} & AY \end{array}$$

Наша реализация

1. Метод `execute` не принимает параметров - все данные передаются только через конструктор

Наша реализация

1. Метод `execute` не принимает параметров - все данные передаются только через конструктор
2. Не возвращать никаких значений

Наша реализация

1. Метод `execute` не принимает параметров - все данные передаются только через конструктор
2. Не возвращать никаких значений
3. Все реализации должны выбрасывать одно и тоже исключение `CommandException`

Почему так? Напишите в чат или скажите голосом.

Задание

1. У танка ограничен запас по топливу. Танк может двигаться, только если у него еще не кончилось топливо.

Задание

1. У танка ограничен запас по топливу. Танк может двигаться, только если у него еще не кончилось топливо.
2. Часть танков имеют запас по топливу, а часть нет.



Тайминг: 20 мин

Макрокоманда

```
class MacroCommand: public Command {
    Command *commands;
    int comCount;
public:
    MacroCommand(Command *comArray, int elemCount) {
        commands = comArray;
        comCount = elemCount;
    }
    void execute(int a, string& b) {
        for (int i = 0; i < comCount; i++) {
            commands[i]->execute(a, b);
        }
    }
}
```

Особенности реализации макрокоманды

Передача аргументов

Особенности реализации макрокоманды

Передача аргументов

Возврат значения

Особенности реализации макрокоманды

Передача аргументов

Возврат значения

Обработка исключений

Наша реализация макрокоманды

Как эти три вещи реализуются у нас?

Задание

1. Реализовать команду Поворот объекта вокруг собственной оси.



Тайминг: 20 мин

Задание

Подумайте над реализацией следующих задач:

1. Как будет выглядеть одновременное выполнение нескольких танковых боев?
2. Как обеспечить циклическое выполнение какого-либо действия?



Тайминг: 10 мин

Ответьте на вопросы

1 Назовите три вещи, которые больше всего запомнились про паттерн Команда

2 Где Вы можете применить команду на практике?


Одна мысль на
лайде

без картинок

Использование
цифр для опроса

0-9





Спасибо за внимание!
Приходите на следующие вебинары

фото

Тюменцев Евгений

Генеральный директор

HWdTech LLC

etyumentcev@gmail.com