|  |  |
| --- | --- |
| C:\Users\7636~1\AppData\Local\Temp\Rar$DRa0.802\герб_синий.jpg | |
| МИНОБРАНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **МИРЭА – Российский технологический университет** | |
| Институт информационных технологий  Кафедра математического обеспечения и стандартизации информационных технологий | |
| **Отчет по проекту «Наш пинг-понг»**  По дисциплине: СиП инженерия | |
|  | Учебная группа: ИВБО-02-16  Студент: Суляев А.Ю.  Преподаватель: Миронов Антон Николаевич |
| Москва 2018 | |

Содержание

[Постановка задачи 2](#_Toc530930382)

[Теория 3](#_Toc530930383)

[Код 4](#_Toc530930384)

[Описание методов(функций) 6](#_Toc530930385)

[Таблица переменных 6](#_Toc530930386)

[Статистика команды с github.com 6](#_Toc530930387)

# Постановка задачи

При помощи ООП описать класс платформа.

**Поля**:

Время последнего изменения

Текущая ширина поля

Скорость дощечки

Текущие координаты

Ширина дощечки

**Методы**:

Запрос информации о ширине поля у класса поле

Перемещение платформы

# Теория

**Объе́ктно-ориенти́рованное программи́рование (ООП)** — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Идеологически ООП — подход к программированию как к моделированию информационных объектов, решающий на новом уровне основную задачу структурного программирования: структурирование информации с точки зрения управляемости, что существенно улучшает управляемость самим процессом моделирования, что, в свою очередь, особенно важно при реализации крупных проектов.

Управляемость для иерархических систем предполагает минимизацию избыточности данных (аналогичную нормализации) и их целостность, поэтому созданное удобно управляемым — будет и удобно пониматься. Таким образом, через тактическую задачу управляемости решается стратегическая задача — транслировать понимание задачи программистом в наиболее удобную для дальнейшего использования форму.

Основные [принципы](https://wiki2.org/ru/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF) структурирования в случае ООП связаны с различными аспектами базового понимания предметной задачи, которое требуется для оптимального управления соответствующей моделью:

* [абстрагирование](https://wiki2.org/ru/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) для выделения в моделируемом предмете важного для решения конкретной задачи по предмету, в конечном счёте — контекстное понимание предмета, формализуемое в виде класса;
* [инкапсуляция](https://wiki2.org/ru/%D0%98%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F) для быстрой и безопасной организации собственно иерархической управляемости: чтобы было достаточно простой команды «что делать», без одновременного уточнения как именно делать, так как это уже другой уровень управления;
* [наследование](https://wiki2.org/ru/%D0%9D%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) для быстрой и безопасной организации родственных понятий: чтобы было достаточно на каждом иерархическом шаге учитывать только изменения, не дублируя всё остальное, учтённое на предыдущих шагах;
* [полиморфизм](https://wiki2.org/ru/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC) для определения точки, в которой единое управление лучше распараллелить или наоборот — собрать воедино.

То есть фактически речь идёт о прогрессирующей организации информации согласно первичным семантическим критериям: «важное/неважное», «ключевое/подробности», «родительское/дочернее», «единое/множественное». Прогрессирование, в частности, на последнем этапе даёт возможность перехода на следующий уровень детализации, что замыкает общий процесс.

# Код



|  |  |
| --- | --- |
|  | last\_time\_change = current\_time; // обновление времени |
|  | } |
|  | } |

# Описание методов(функций)

Запрос информации о ширине поля у класса поле

get\_width();

Перемещение платформы с учётом скорости, проверки на движение, времени.

UpdatePhysicsPlatform()

# Таблица переменных

|  |  |
| --- | --- |
| int widthField | Ширина поля |
| time\_last\_change | Время последнего изменения |
| Current\_width | Текущая ширина поля |
| speed | Скорость дощечки |
| current\_pos | Текущие координаты |
| platform\_width | Ширина дощечки |

# Статистика команды с github.com

