Кирилл Волков @ MERA
github.com/vulko/Cpp_Basics_Lectures

C++

Основные преимущества ООП.
Наследование, инкапсуляция, полиморфизм:
Абстрактные классы, интерфейсы, виртуальные функции.

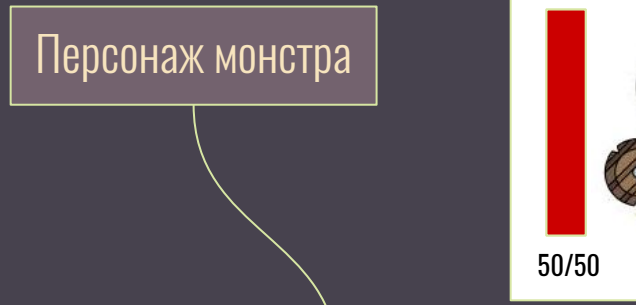

ООП
Инкапсуляция
“Сокрытие” реализации
public, private, protected
Наследование
Иерархия
Распределение функционала
Перегрузка функций
Полиморфизм
Абстракция
Виртуальность
Перегрузка

Абстракция объекта

Абстракция конкретного объекта объекта

Абстракция объекта

Абстракция конкретного объекта объекта



Имплементация объекта

```
class Player : public GameObject {

private:
    int posX, posY;          // inherited
    string mName;            // inherited

   // health
   int mMaxHealth;
   int mCurrentHealth;

   // damage
   int mDamage;

   // exp
   int mExpirience;
   int mLevel;

}
```

Абстракция конкретного объекта объекта

Персонаж игрока

Свойства

Имя
Положение в игровом мире
Очки здоровья, максимальные и текущие
Урон
Опыт/Уровень
Артефакты или модификаторы урона/здоровья

Абстракция объекта

Абстракция конкретного объекта объекта

Функционал

Ходить по игровому миру в соотв. с действиями пользователя
Изменение состояния игрока после битвы, после отдыха
Взаимодействие с игровым миром - битва, отдых
Получение опыта

Имплементация объекта

```
class Player : public GameObject {

public:
    void move(int x, int y);        // inherited
    void draw();                    // inherited

    void takeDamage(int dmg) {
        mCurrentHealth -= dmg;
        if (mCurrentHealth <= 0) {
            // player is dead
        }
    }

    void getLevel() {
        mLevel = mExperience / EXP_PER_LVL;
    }

    void addExp(int exp) { mExpirience += exp; }

}
```

Абстракция конкретного объекта объекта

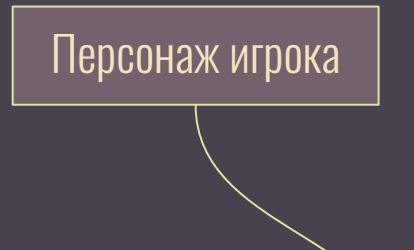
Персонаж игрока

Функционал

Ходить по игровому миру в соотв. с действиями пользователя
Изменение состояния игрока после битвы, после отдыха
Взаимодействие с игровым миром - битва, отдых
Получение опыта

Абстракция объекта

Абстракция конкретного объекта объекта



Внутренние взаимодействия

Изменения состояния игрока в процессе битвы.

Получение опыта -> новый уровень -> изменение показателей очков жизни и урона.

Абстракция объекта

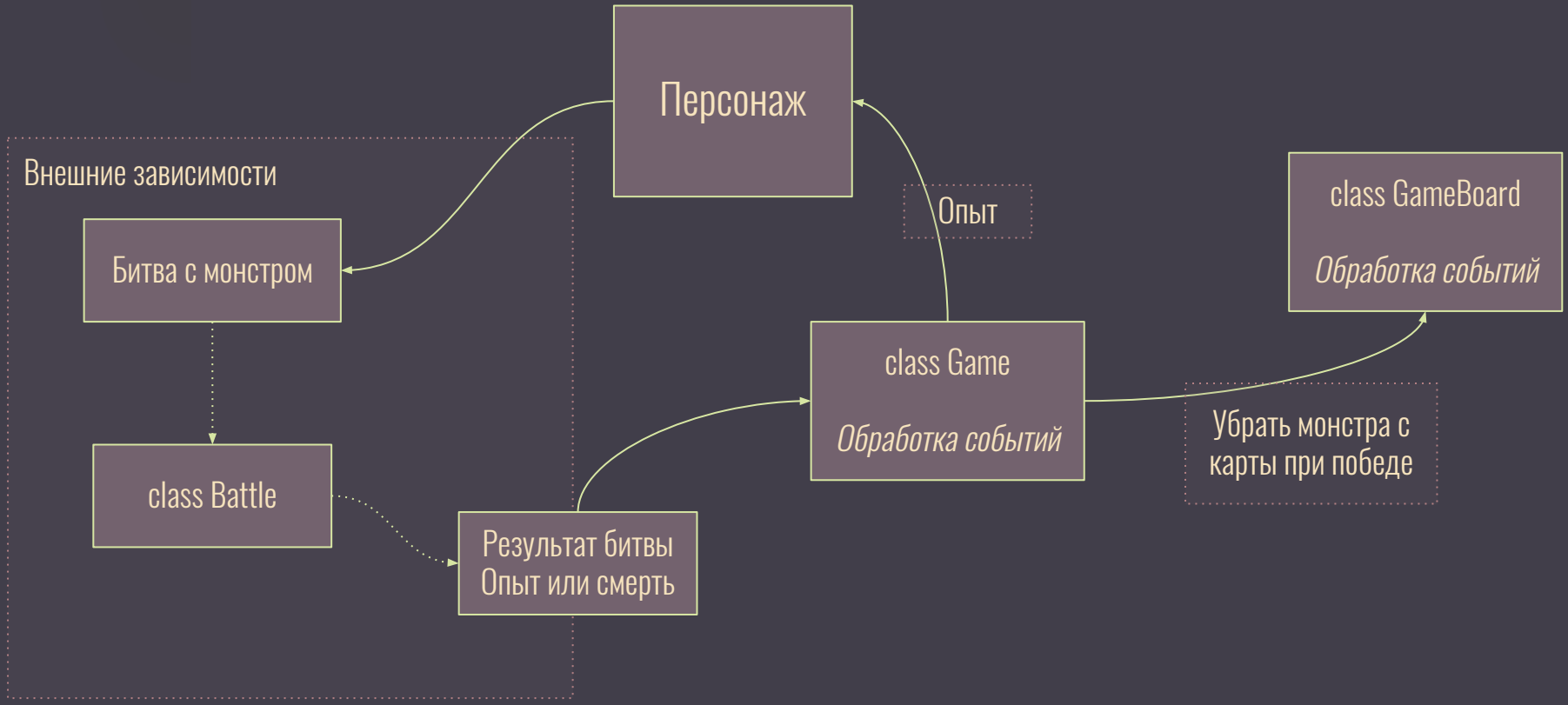
Абстракция конкретного объекта объекта

Внешние взаимодействия

Игровой мир: перемещение, взаимодействие с другими объектами

Битва с монстром

Распределение функционала (decoupling)



Распределение функционала (decoupling)

Преимущества такого подхода:

Логика работы распределяется между отдельными объектами, каждый из которых занимается “своими задачами”

Легче найти причину неправильной работы

Легче тестировать -> Unit testing и Integration testing

Лучше структурированность и читабельность кода

Легче модифицировать и расширять функционал