



HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Лекция 1

Основы процедурного программирования (часть 1)

Программирование на языке Java

Роман Гуров

ВШЭ БИ 2021





Почему программирование?

- Широко востребовано => прибыльно
- ERP разработки и заинтересованность бизнесов в них (1С)
- Даже если вы не пойдете работать по профилю, но пойдете в IT-сегмент, вам будет проще понимать чем заняты разработчики. Есть доступ к различным профессиям, например, PM: код писать не придется, но понимать не помешает =)



Цель занятия

- Изучить принципы алгоритмического мышления
- Абстрактно рассмотреть основные конструкции, встречающиеся в программировании
- Подготовиться к комфортному освоению этих концепций на языке Java

- Если вдруг возникнет вопрос, даже глупый/запоздалый, не стесняйтесь спрашивать голосом или писать в чат. Перебивать можно :)

Что такое программирование

- Это написание текста
- Текст программы состоит из инструкций, которые выполняются при запуске программы

```
#ifndef SIMPLE_CHESS_ENGINE_FIGURE_H
#define SIMPLE_CHESS_ENGINE_FIGURE_H

#include <list>

#include "field.h"
#include "move.h"

class figure {
public:
    enum color {
        e_white,
        e_black
    };

    figure (field::cell cell, color the_color, const field* the_field) noexcept;

    [[nodiscard]] field::cell get_cell () const noexcept;
    [[nodiscard]] color get_color () const noexcept;

    [[nodiscard]] virtual std::list<const move*> get_possible_moves () const noexcept = 0;
    [[nodiscard]] virtual std::string to_string () const noexcept = 0;

protected:
    field::cell _cell;
    const color _color;
    const field* const _field;
};
```



Что такое инструкция

Задача:

Дмитрий хочет поиграть в футбол. Какую последовательность действий ему требуется выполнить?

1. Одеться
2. Взять мяч
3. Выйти на улицу
4. Поставить мяч на поле
5. Разогнаться и пнуть мяч в ворота

Каждая из этих операций является *инструкцией* или, иными словами, *действием* или *операцией*

Что такое программирование

- Это написание текста
- Текст программы состоит из инструкций, которые выполняются при запуске программы
- Сама программа может содержать несколько текстовых файлов, и не только текстовых, вообще.
- Задача программы: обработать входные данные и предоставить выходные данные.
- Текст превращается в программу благодаря специальной программе, которая называется компилятор.

```
#ifndef SIMPLE_CHESS_ENGINE_FIGURE_H
#define SIMPLE_CHESS_ENGINE_FIGURE_H

#include <list>

#include "field.h"
#include "move.h"

class figure {
public:
    enum color {
        e_white,
        e_black
    };

    figure (field::cell cell, color the_color, const field* the_field) noexcept;

    [[nodiscard]] field::cell get_cell () const noexcept;
    [[nodiscard]] color get_color () const noexcept;

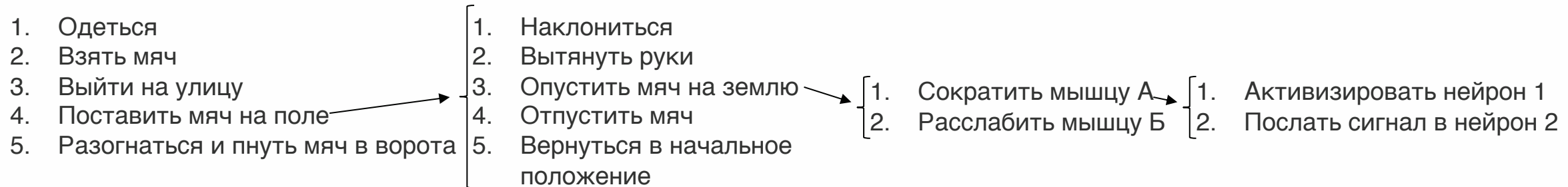
    [[nodiscard]] virtual std::list<const move*> get_possible_moves () const noexcept = 0;
    [[nodiscard]] virtual std::string to_string () const noexcept = 0;

protected:
    field::cell _cell;
    const color _color;
    const field* const _field;
};
```

Что такое инструкция

Задача:

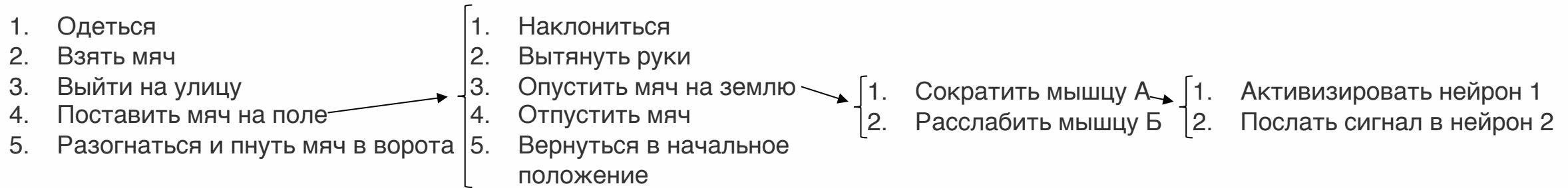
Дмитрий хочет поиграть в футбол. Какую последовательность действий ему требуется выполнить?



Каждая из этих операций является *инструкцией* или, иными словами, *действием* или *операцией*

Казалось бы, задача была решена, но каждый раз решение можно было расписать подробнее. Нельзя было точно утверждать, что программа закончена и готова к выполнению. Почему?

Что такое инструкция



Мы начали составлять программу *сверху*, с самых общих инструкций. После, стали спускаться вниз, разбивая каждую инструкцию подробнее на более маленькие.

Начало – потолок – мы вольны выбрать для себя сами. Но где же пол?

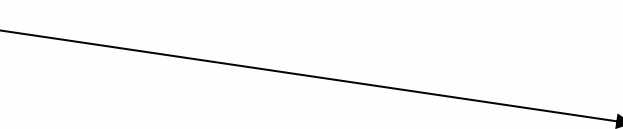
Что такое инструкция

Рассмотрим ещё одну задачу:

Пётр решает номер на контрольной по физике. Для получения ответа ему нужно подставить в итоговую формулу все численные данные и посчитать результат на калькуляторе. Какую последовательность нажатий кнопок ему нужно для этого совершить?

Пусть, итоговая формула такая: $\frac{a+b}{c}$

1. Ввести число a
2. Нажать “+”
3. Ввести число b
4. Нажать “=”
5. Нажать “/”
6. Ввести число c
7. Нажать “=”

- 
1. Нажать “1”
 2. Нажать “3”
 3. Нажать “7”
 4. ...
 5. ...

- Этот алгоритм получился законченным, мы не можем разбивать действия дальше
- В контексте этой задачи не предусмотрены действия, кроме нажатий на кнопки
- Мы обошлись только нажатиями, а значит уверены, что неоднозначностей нет



Что такое инструкция

Так где же всё-таки пол?

- В задаче с калькулятором нам заранее известен набор инструкций, из последовательности которых должна состоять программа
- В задаче с мячом такого набора не дано, можно лишь предположить, что человек, исполняющий написанные инструкции, поймёт каждую из них

В реальном программировании всегда есть какой-то набор уже заданных инструкций, которыми можно пользоваться, и каждая программа в конечном итоге сводится к этому набору

Нельзя просто написать в коде программы “сделай_мне_красивый_сайт”, всё создание сайта нужно делать самостоятельно, пользуясь инструкциями, предоставленными языком программирования (ЯП)



Процедурное программирование

Но было бы неудобно, если бы весь код выглядел, как список стандартных инструкций ЯП

Нам казалось естественным и удобным разбить программу на маленькие самостоятельные подпрограммы

поиграть_в_футбол:

1. Одеться
2. Взять мяч
3. Выйти на улицу
4. Поставить мяч на поле
5. Разогнаться и пнуть мяч в ворота

поставить_мяч_на_поле:

1. Наклониться
2. Вытянуть руки
3. Опустить мяч на землю
4. Отпустить мяч
5. Вернуться в начальное положение

Таким образом мы создаем новые инструкции, которыми можем далее пользоваться

В программировании такие подпрограммы называются *процедурами*, причём одна процедура является *главной*, именно с неё начинается выполнение программы

Использование процедур позволяет переиспользовать один алгоритм, не прибегая к повторению его кода



Выражения

Ни один язык программирования не обходится без поддержки математических выражений

Выражение представляет собой набор операторов – сложения, умножения, деления и т.д. – применяемых к некоторым значениям

Например, $\frac{10x^2+7x-3}{42}$ может быть записано в виде выражения как $(10*x*x + 7*x - 3) / 42$

Посмотрим на выражения в деле в Wolfram Alpha

Приоритет операторов в выражениях

Так же, как и в простой математике, в выражениях в программировании существует приоритет между операторами

Больший приоритет	*, /
Меньший приоритет	+, -

При прочих равных, сначала выполнятся операторы с большим приоритетом; для операций равного приоритета выполнение идёт слева направо

$$\begin{array}{l} 10 \quad x \\ \downarrow \times \\ 10 * x \quad x \quad 7 \quad x \\ \downarrow \times \quad \downarrow \times \\ 10 * x * x \quad 7 * x \\ \downarrow + \\ 10 * x * x + 7 * x \quad 3 \\ \downarrow - \\ 10 * x * x + 7 * x - 3 \quad 42 \\ \downarrow \div \\ (10 * x * x + 7 * x - 3) / 42 \end{array}$$

$$\begin{array}{l} 10 \quad x \\ \downarrow \times \\ 10 * x \quad x \quad 7 \quad x \\ \downarrow \times \quad \downarrow \times \\ 10 * x * x \quad 7 * x \quad 3 \quad 42 \\ \downarrow + \quad \downarrow \div \\ 10 * x * x + 7 * x \quad 3 / 42 \\ \downarrow - \\ 10 * x * x + 7 * x - 3 / 42 \end{array}$$



Пример использования выражения

Пока что мы просто рассмотрели концепцию выражения, но никак не связали их с самой программой

Самое простое действие программы – вывести текст на экран

Допустим, мы хотим напечатать на экране результат выражения $(5 + 3) * 7$

Предположительно, код должен выглядеть как-то так:

```
print (5 + 3) * 7
```

И, сюрприз, одна эта строчка уже является полноценной программой на языке Python!



Переменная

Что, если хотим посчитать выражение $\frac{10x^2+7x-3}{42}$, но при этом вместо x подставить $y^2 - y + 2$?

Если сразу подставить это значение выражение вместо каждого x , то его придется повторять трижды. Хотелось бы посчитать значение x один раз, куда-нибудь записать и в дальнейшем читать его оттуда

Переменная – по своей сути является некоторой ячейкой в камере хранения, в которую программа может положить объект и потом обращаться к нему

Для записи объекта в переменную используется оператор присваивания (=):

$y = 5$

Оператор присваивания выполняет конкретное действие – присваивает переменной указанное значение.

Это не утверждение того, что y равен пяти, а именно запись того, что справа, в то, что слева

Переменная

Что, если хотим посчитать выражение $\frac{10x^2+7x-3}{42}$, но при этом вместо x подставить $y^2 - y + 2$?

```
y = 5
x = y*y - y + 2
answer = (10*x*x + 7*x - 3) / 42
print answer
```

Переменная называется переменной, потому что её значение можно изменять:

```
y = 5
x = y*y
x = x - y
x = x + 2
answer = (10*x*x + 7*x - 3) / 42
print answer
```

==

```
y = 5
x = y*y
x -= y
x += 2
answer = (10*x*x + 7*x - 3) / 42
print answer
```


Ветвление

Далеко не все алгоритмы работают прямолинейно



Задача:
Перейти дорогу.

Простого указания пересечь дорогу недостаточно, ведь можно попасть под машину

Сначала надо убедиться, что машин нет, и только в этом случае начинать движение



Ветвление

Задача:

Перейти дорогу.

Пусть уже существуют процедуры:

перейти_дорогу

посмотреть_направо -> да/нет

посмотреть_налево -> да/нет

При этом процедура `посмотреть_*` говорит, есть ли машина в том направлении

```
есть_машина_справа = посмотреть_направо()
```

```
есть_машина_слева = посмотреть_налево()
```

```
если не есть_машина_справа и не есть_машина_слева то:  
    перейти_дорогу()
```

иначе:

```
    пойти_домой()???
```



Ветвление

В общем виде конструкция условного ветвления выглядит так:

```
если <условие> то:  
    <действие 1>  
иначе:  
    <действие 2>
```

Исполнение в таком случае расходится на две возможные ветви: либо выполнится действие 1, либо действие 2, но оба никогда.



Ветвление

Задача:

Найти количество корней квадратного уравнения $ax^2 + bx + c = 0$, где a , b и c даны

```
a = считать_число()
```

```
b = считать_число()
```

```
c = считать_число()
```

```
D = b*b - 4*a*c
```

```
если D > 0:
```

```
    напечатать "Два корня"
```

```
иначеесли D == 0:
```

```
    напечатать "Один корень"
```

```
иначе:
```

```
    напечатать "Корней нет"
```

```
напечатать "Задача решена"
```

Когда альтернатив становится больше, чем две,
можно задать их таким образом

По-английски такая структура называется if-elif-else