Алгоритмы и Алгоритмические Языки (C++)

Семинар #4:

1. Классы в C++: методы, модификаторы доступа;
2. Проверка инварианта структуры данных;
3. Конструкторы и деструктор;
4. Конструктор и оператор “=”: `copy/move`-семантика;
5. Обработка ошибок в C++: `throw + try-catch`.

06.02.2024

HSE University

Классы в C++: методы, модификаторы доступа

Классы и модификаторы доступа

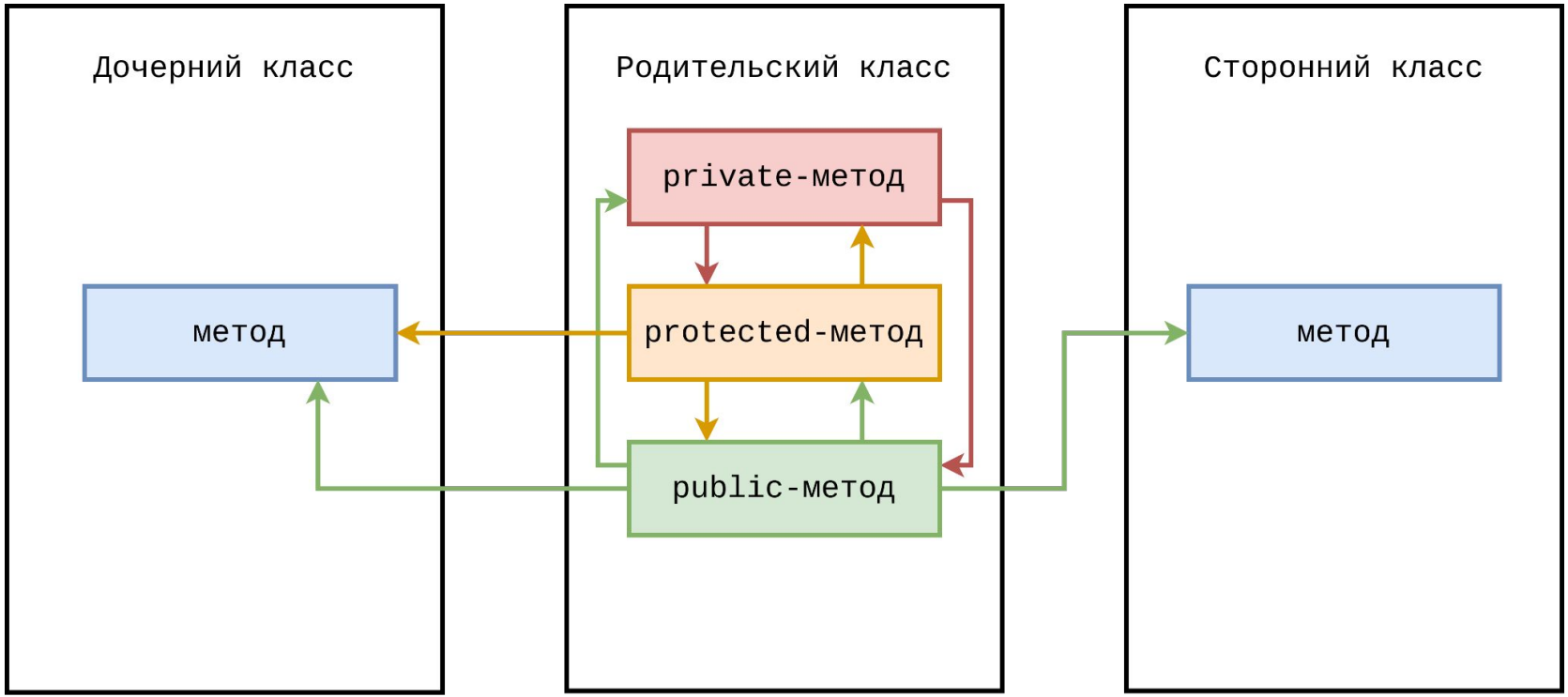
Модификаторы доступа:

- `private` – поля и методы доступны только другим методам;
- `public` – поля и методы доступны всем;
- `protected` – поля и методы доступны наследниками.

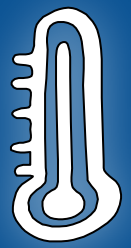
Отличие struct от class:

- По умолчанию для `struct` – всё `public`;
- По умолчанию для `class` – всё `private`;

Классы и модификаторы доступа



Проверка инвариантов структуры данных



Инвариант структуры данных

- Является функцией от полей структуры;
- Обозначает владение освобождаемыми ресурсами;
- Может иметь значения `true/false`;
  - `true` – структура данных корректна и готова к работе;
  - `false` – структура данных непригодна к работе.

Примеры:

```
struct Stack
{
    data_t* array;
    size_t size;
    size_t capacity;
};
```

```
struct Matrix
{
    double** data;
    size_t size_x;
    size_t size_y;
};
```

```
struct Node {
    Node* next;
    data_t data;
};

struct List {
    struct Node* head;
};
```

Инвариант структуры данных

```
struct Stack
{
    data_t* array;
    size_t size;
    size_t capacity;
};
```

```
struct Matrix
{
    double** data;
    size_t size_x;
    size_t size_y;
};
```

```
struct Node {
    Node* next;
    data_t data;
};

struct List {
    struct Node* head;
};
```

Проблема C: при инициализации поле имеет любое значение.

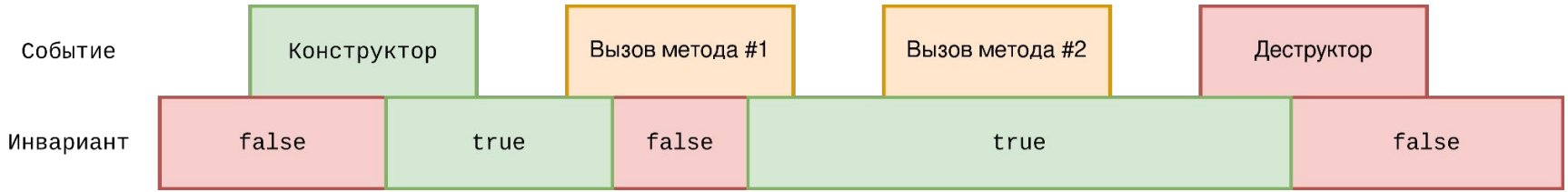
Решение в языке C:

- В начале работы: `data_structure_init(&data_structure);`
- В конце работы: `data_structure_free(&data_structure);`

Проблемы: легко забыть `init/free`, `double free`, `double init`, ...

Конструкторы и деструктор

Конструкторы и деструктор



Конструктор: аллоцирует ресурсы;

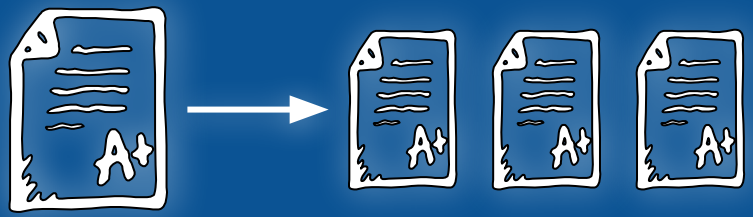
Деструктор: освобождает ресурсы;

Методы:

- Могут временно приводить структуру в “плохое” состояние;
- В конце работы должны гарантированно вернуть “хорошее”;

HSE University

Конструктор копирования, оператор присваивания



HSE University

Обработка ошибок в C/C++: `throw` + `try-catch`

Вопросы?

Красивые иконки взяты с сайта handdrawngoods.com