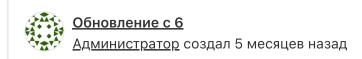


Развилка из недоступного проекта



Имя	Последний коммит	Последнее обновление
□ образцы кода	Инициировать фиксацию	1 год назад
<b>□</b> <u>образцы данных</u>	Инициировать фиксацию	1 год назад
<b>□</b> <u>наборы данных</u>	Инициировать фиксацию	1 год назад
<b>т</b> материалы	Обновление с 6	5 месяцев назад
<b>р</b> азное	Инициировать фиксацию	1 год назад
□ источник	Инициировать фиксацию	1 год назад
◆ .gitignore	Обновление с 4	1 год назад
<u> МЗМЕНЕНИЙ</u>	<u>Инициировать фиксацию</u>	1 год назад
<mark>₽ ЛИЦЕНЗИЯ</mark>	<u>Инициировать фиксацию</u>	1 год назад
M♣ README.md	<u>Обновление с 5</u>	7 месяцев назад

## README.md

# День 08. Учебный лагерь

# Приключение, опасность и какао

# Содержание

- 1. <u>Глава I</u>
  - 1.1. Основные правила
- 2. <u>Глава II</u>
  - 2.1. Правила дня
- 3. <u>Глава III</u>
  - 3.1. вступление
- 4. <u>Глава IV</u>
  - 4.1. Упражнение 00: Арифметика
- 5. <u>Глава V</u>
  - 5.1. <u>Упражнение 01. Ботаника</u>
- 6. <u>Глава VI</u>
  - 6.1. Упражнение 02. Горячий шоколад

# Глава I

### Основные правила

- Ваши программы не должны завершаться неожиданно (выдавая ошибку при правильном вводе). Если это произойдет, ваш проект будет считаться неработоспособным и получит 0 баллов при оценке.
- Мы рекомендуем вам создать тестовые программы для вашего проекта, даже если эту работу не нужно будет отправлять и она не будет оцениваться. Это даст вам возможность легко проверить свою работу и работу ваших коллег. Вы найдете эти тесты особенно полезными во время вашей защиты. Действительно, во время защиты вы можете использовать свои тесты и/ или тесты коллеги, которого вы оцениваете.
- Отправьте свою работу в назначенный репозиторий git. Оцениваться будет только работа в репозитории git.

• Если ваш код использует внешние зависимости, он должен использовать модули Go для управления ими.

#### Глава II

### Правила дня

- Вы должны только сдать \*.go файлы и (в случае внешних зависимостей) go.mod + go.sum
- Ваш код для этой задачи должен быть собран с помощью всего лишь go build

#### Глава III

#### вступление

Люди склонны говорить, что одно из основных различий между Go и C — безопасность указателей. Отчасти это правда, и Go будет очень стараться не позволить вам выстрелить себе в ногу, играя с указателями. Но мы уже достаточно далеко в джунглях, чтобы можно было немного поиграть с опасностью, тебе не кажется?

#### Глава IV

#### Упражнение 00: Арифметика

Здесь, в джунглях, вы можете встретить странных существ, с которыми нужно обращаться необычным способом. Для этой задачи вам нужно написать функцию getElement(arr []int, idx int) (int, error), которая принимает и индекс и возвращает вам элемент с этим индексом. Кажется достаточно простым, а? Но вот одно условие - вы не можете использовать поиск по этому индексу (например arr[idx]), разрешен только поиск по первому элементу (arr[0]). Возможно, вам придется вспомнить немного C, чтобы выполнить это упражнение.

В случае любого недопустимого ввода (пустой срез, отрицательный индекс, индекс за пределами) функция должна вернуть ошибку с текстовым объяснением проблемы.

#### Глава V

### Упражнение 01. Ботаника

Вам повезло! Вы нашли довольно редкие растения:

```
type UnknownPlant struct {
    FlowerType string
    LeafType string
    Color int `color_scheme:"rgb"`
}

type AnotherUnknownPlant struct {
    FlowerColor int
    LeafType string
    Height int `unit:"inches"`
}
```

Ну да, нынешнее представление немного беспорядок. Ваша цель состоит в том, чтобы написать одну функцию describePlant, которая будет принимать любые типы растений (да, она должна работать со структурами разных типов), а затем распечатать все поля в виде пар ключ-значение, разделенных запятой (обратите внимание на теги), например: :

```
FlowerColor:10
LeafType:lanceolate
Height(unit=inches):15
```

### Глава VI

### Упражнение 02. Горячий шоколад

Ладно, теперь пора расслабиться и выпить какао. Какао обычно поставляется в упаковках (см. прилагаемый zip-архив). Вам не нужно каким-либо образом изменять код в упакованных файлах, единственное, что вам нужно сделать, это написать код (включая файлы какао как часть вашего проекта), который создаст пустое окно графического интерфейса Mac OS по умолчанию (размер 300х200) с названием «Школа 21». Это проще, чем вы думаете!