 [Как найти доходное Дело, дающее](#) душевное удовлетворение? Запишись и узнай это на семинаре онлайн бесплатно!

Есть противопоказания. Посоветуйтесь с врачом.

[Как удвоить заработок!](#) Уникальное открытие психологов: Технология Исполнения Желаний

[Яндекс.Директ](#) [Все объявления](#)

Разработка простого генетического алгоритма



Открытый Helgus-мастер-Класс — Н-м-К

Это незавершённая статья из области эвентологии и её применений, редактируемая при участии Мастера

Оглавление [\[Скрыть\]](#)

- 1 1. Теоретическая часть (описание ГА и его операторов)
 - 1.1 Эволюционные алгоритмы
 - 1.1.1 Обоснование для имитации процессов эволюции
 - 1.1.2 Сопоставление природных и компьютерных понятий:
 - 1.2 Обобщенный эволюционный алгоритм:
 - 1.3 ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ
 - 1.4 Простой ГА
- 2 Кодирование решений в ГА
 - 2.1 Бинарное кодирование
 - 2.2 Грэй-код в ГА
 - 2.3 Динамическое кодирование параметров (ДКП)
- 3 Селекция в ГА
 - 3.1 Пропорциональная селекция
 - 3.2 Ранговая селекция
 - 3.3 Турнирная селекция
 - 3.4 Элитарная селекция
- 4 Скрещивание в ГА
- 5 Мутация в ГА
- 6 2. Практическая часть (описание программы и порядка работы с ней)
 - 6.1 Порядок работы с программой

1. Теоретическая часть (описание ГА и его операторов)

[\[редактировать\]](#)

Эволюционные алгоритмы

[\[редактировать\]](#)

Эволюционные алгоритмы являются оптимизационными и поисковыми процедурами, имитирующими процессы естественной селекции.

По тому, какие структуры используются в оптимизации, какие стратегии репродукции используются, какие генетические операции задействуются и т.п., эволюционные алгоритмы могут быть подразделены на

- Эволюционное программирование
- Эволюционное программирование
- Классифицирующие системы
- Генетические алгоритмы

- Генетическое программирование

Обоснование для имитации процессов эволюции

[\[редактировать\]](#)

- Эволюционные алгоритмы как наука:

Являются средством для проверки гипотез (например, моделей) в **биологии**, **социологии**, религии, и т.д., например, ЕА могли бы дать ответы на следующие вопросы:

1. Почему **хромосомы** гомологичны?
2. Каковы преимущества полового размножения?
3. Каковы функции невыраженных **генов** и кодонов?
4. Для чего нужна миграция?
5. Каковы преимущества моногамии?

- Эволюционные алгоритмы как техника:

- Оптимизация функций
- Комбинаторная оптимизация
- Автоматическое обучение
- Решение задач

Сопоставление природных и компьютерных понятий:

[\[редактировать\]](#)

| | |
|-----------------------|--|
| Природа | Компьютер |
| Индивид | Решение задачи |
| Популяция | Множество решений |
| Пригодность | Качество решения |
| Хромосома | Представление решения (например, множество параметров) |
| Ген | Часть представления решения (например, параметр или степень свободы) |
| Рост | Декодирование представления решения |
| Скрещивание | Оператор поиска |
| Мутация | Оператор поиска |
| Естественная селекция | Повторное использование хороших решений |

Обобщенный эволюционный алгоритм:

[\[редактировать\]](#)

1. Инициализировать популяцию
2. Оценить популяцию
3. Повторять, пока не выполнится условие останова
 1. *Селекция*. Отобрать часть популяции для воспроизводства.
 2. *Рекомбинация*. Выполнить скрещивание "генов" отобранных родителей.
 3. *Мутация*. Случайным образом осуществить мутацию полученной популяции.
 4. *Оценивание*. Оценить пригодность популяции.
 5. На основе полученной пригодности выбрать выживших индивидов.

Замечание 1. Не во всех эволюционных алгоритмах представлены все шаги.

Замечание 2. Количество различных эволюционных алгоритмов совпадает с количеством исследователей, работающих в данной области!

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

[\[редактировать\]](#)

- В генетических алгоритмах (ГА) существует строгое различие между фенотипом (т.е. взрослым индивидом, решением) и генотипом (т.е. хромосомой, представлением решения)

- ГА работают, аналогично природе, только с генотипом, поэтому требуется процесс декодирования генотипа в фенотип и обратно («обобщенный» рост)
- Хромосомы являются строками символов (например, 0 и 1)
- Взрослые индивиды могут быть чем угодно, если только существует способ их кодирования-декодирования с использованием строк символов (хромосом)
- Примерами фенотипов являются векторы параметров или списки альтернатив

[\[редактировать\]](#)

Простой ГА

1. Генерировать популяцию хромосом случайным образом
2. Декодировать каждую хромосому для получения индивидов
3. Оценить пригодность каждого индивида
4. Генерировать новую популяцию путем клонирования (копирования), частичной рекомбинации и частичной мутации хромосом наиболее пригодных индивидов
5. Повторять шаги 2, 3 и 4 до тех пор, пока не будет выполнено условие останова

Замечание 1. Поколения не перекрываются (т.е. существуют по отдельности)

Замечание 2. Клонирование используется для имитации выживания родителей в более, чем одном поколении

Кодирование решений в ГА

[\[редактировать\]](#)

Бинарное кодирование

[\[редактировать\]](#)

Целочисленные параметры

- Если p изменяется в пределах $\{0, 1, \dots, 2^N - 1\}$, то мы кодируем его с использованием бинарного представления $e = [b_0, \dots, b_N]$ с N битами $b_i \in \{0, 1\}$ так, что $p = \sum_{i=1}^N b_i * 2^{i-1}$.
- Если $p \in \{M, M + 1, \dots, M + 2^N - 1\}$, то кодируем $p - M$ тем же самым способом.
- Если $p \in \{0, 1, \dots, L - 1\}$ и нет такого N , что $L = 2^N$, то возникает проблема.

Возможные решения проблемы:

1. Выбрать $N = \log_2(L) + 1$ и полагать любую строку $[b_0, \dots, b_N]_{10} \geq L$ равной $p = L - 1$. *Проблема:* каждому p между 0 и $L - 2$ соответствует единственная бинарная строка, в то время как $p = L - 1$ представлен $2^N - L$ строками, т.е. представление отдает предпочтение этому параметру.
2. Выбрать $N = \log_2(L) + 1$ и отражать каждое десятичное число $e_{10} = [b_0, \dots, b_N]_{10}$ в $p = \left\lfloor \frac{e_{10}}{2^N - 1} * (L - 1) \right\rfloor$ (масштабирование). *Проблема:* некоторые из значений p будут представлены двумя бинарными строками, т.е. также имеем предпочтение, хотя и намного более слабое, чем в первом случае.

Вещественные параметры могут быть представлены числами с фиксированной точкой или целыми числами путем масштабирования и дискретизации.

- Если $p \in \{p_{min}, p_{max}\}$, то e является бинарным представлением целой части числа, определяемого как $(2^N - 1) * \frac{p}{(p_{max} - p_{min})}$.
- Для вещественных параметров имеет место конфликт между желанием иметь как можно более короткий ген для обеспечения хорошей сходимости и необходимостью получить результат с определенной точностью.

Для кодирования параметров, которые могут принимать только *малое число значений* может быть использован не бинарный алфавит. Буквы такого алфавита представляются как отдельные неделимые гены (атомы). Например, если параметр $p \in \{-1, 0, 1\}$, то мы можем использовать кодирование $e \in \{a, b, c\}$, а если $p \in \{Rim, Amsterdam, London, New-York, Pariz, Krasnoarsk, Kemerovo\}$, то $e \in \{R, A, L, N, P, Kr, Ke\}$.

Иногда и целочисленные или *вещественные параметры рассматриваются как атомы* (символы очень большого алфавита).

Векторы параметров кодируются путем комбинирования кодов $[b_0^i, \dots, b_N^i]$ каждой компоненты

- коды множества параметров могут быть просто поставлены один за другим (например, $[b_0^1, \dots, b_N^1, \dots, b_0^M, \dots, b_N^M]$ или встроены один в другой (например, $[b_0^1, \dots, b_0^M, \dots, b_N^1, \dots, b_N^M]$).

- порядок параметров в векторе является исключительно важным, особенно при первом способе кодирования.

[\[редактировать\]](#)

Грэй-код в ГА

- бинарные коды параметров часто получают применением кодирования Грэя
- код Грэя представляет каждое число последовательности целых чисел $0, 1, \dots, 2^N - 1$ в виде бинарной строки длины N
- смежные числа в коде Грэя отличаются только на один бит
- существует множество кодов Грэя
- *Алгоритм* (двоично-рефлективный код Грэя): начать со строки, состоящей из нулей и последовательно инвертировать самый правый бит, который может дать новую строку.

Целое Двоичное Код

| число | число | Грея |
|-------|-------|------|
| 0 | 000 | 000 |
| 1 | 001 | 001 |
| 2 | 010 | 011 |
| 3 | 011 | 010 |
| 4 | 100 | 110 |
| 5 | 101 | 111 |
| 6 | 110 | 101 |
| 7 | 111 | 100 |

- Достоинства: случайные инверсии битов производят (с большей вероятностью) небольшие изменения, нет резких скачков в метрике Хэмминга, соседние числа остаются соседними и после бинаризации
- Недостатки: в тех редких случаях, когда изменения являются большими, они значительно больше, чем в бинарном коде, далекие числа могут оказаться соседними после кодирования

Динамическое кодирование параметров (ДКП)

[\[редактировать\]](#)

- ДКП является методом решения проблемы противоречия точность-скорость при кодировании вещественных параметров
- ДКП в процессе решения задачи делает нечто, похожее на следующую последовательность процедур:
 1. Сначала выполнить эксперименты с короткими генами, чтобы быстро получить результат с небольшой точностью
 2. Ограничить поиск областью ранее полученного решения (т.е. увеличить точность) и снова провести эксперименты с короткими генами
 3. Повторять шаг 2 до тех пор, пока не будет получена достаточная точность
- ДКП применяет эту стратегию с помощью накопления статистики сходимости по каждому параметру и задействования *оператора увеличения* (zoom) в случаях, когда популяция ограничивается некоторым субрегионом поискового интервала для параметров.
- Оператор увеличения переключает интерпретацию гена в смысле удвоения точности поиска. Поиск ограничивается целевым подинтервалом .
- Оператор увеличения *не обратим*: его надо применять очень осторожно для того, чтобы исключить преждевременную сходимость

Селекция в ГА

[\[редактировать\]](#)

- Селекция - это оператор, посредством которого индивиды (т.е. хромосомы) выбираются для спаривания и порождения потомков
- Для имитации естественной селекции индивиды с более высокой пригодностью должны выбираться с большей вероятностью
- Существует большое число различных моделей селекции, некоторые из которых не имеют биологических аналогов
- Большинство схем селекций, используемых в ГА, создают промежуточную популяцию и затем выбирают из нее случайным образом пары индивидов для скрещивания
- Предположения:
 1. Мера Q качества решения (целевая функция) задачи известна
 2. Она должна быть максимизирована
 3. Она должна быть всегда положительной
 4. Мы рассматриваем меру Q качества индивида как его пригодность

Пропорциональная селекция

- Это наиболее часто используемый вид селекции в ГА

- Пусть f_i - пригодность индивида i , $\bar{f} = \frac{1}{N} * \sum_i f_i$ - средняя пригодность популяции. Тогда в пропорциональной селекции индивид i выбирается для репродукции с вероятностью $p_i = \frac{f_i}{\sum_j f_j} = \frac{f_i}{\bar{f} * N}$.

- Другая возможность - выбирать случайное вещественное число $r \in \left[0, \sum_{j=1}^N f_j\right]$ и отобрать того индивида, для которого $\sum_{j=1}^{i-1} f_j \leq r < \sum_{j=1}^i f_j$, где $\sum_{j=1}^0 f_j = 0$.

Проблема: такой алгоритм работает относительно медленно.

- В равновесном ГА популяциям не разрешается увеличиваться или сокращаться, поэтому для репродукции отбирают N индивидов. Отсюда следует, что *ожидаемое число копий* каждого индивида в промежуточной популяции равно $N_i = p_i * N = \frac{f_i}{\bar{f}}$.

- N_i обычно является вещественным числом. Реальное число копий (целое) может варьироваться около N_i .
- Индивиды с пригодностью выше средней имеют более одной копии в промежуточной популяции, а индивиды ниже средней пригодности могут не иметь ни одной (в среднем).
- Проблемы с пропорциональной селекцией:

1. *Преждевременная сходимость:*

- Индивид с $f_i \gg \bar{f}$, но $f_i < f_{max}$ был получен на ранних поколениях.
 - Так как $N_i \gg 1$, гены такого индивида довольно быстро распространятся на всю популяцию.
 - В таком случае рекомбинация не может более производить новых индивидов (только мутация может) и $\bar{f} \ll f_{max}$ навсегда.

1. *Стагнация:*

- Ближе к концу работы алгоритма все индивиды могут получить относительно высокую и примерно равную пригодность, т.е. $\forall i f_i \approx f_{max}$.
 - Тогда $N_1 \approx \dots \approx N_N \approx 1$, что приводит к очень маленькому селективному давлению, т.е. наилучшее решение предпочитается лишь немного больше, чем наихудшее.

Замечание: обе проблемы могут быть преодолены применением методики *масштабирования пригодности* (будет объяснено позже).

Ранговая селекция

- Индивиды сортируются (ранжируются) на основе их пригодности таким образом, чтобы $f_i \geq f_j$ для $i > j$.
- Затем каждому индивиду назначается вероятность p_i быть отобранным, взятая из заданного распределения с ограничением

$$\sum_i p_i = 1$$

- Типичные распределения:

1. Линейное: $p_i = a * i + b (a < 0)$.
2. Отрицательное экспоненциальное: $p_i = a * \exp(b * i + c)$. Это эквивалентно назначению первому индивиду вероятности p , второму - p_2 , третьему - p_3 , и т.д.

- Преимущества:

1. Нет преждевременной сходимости, т.к. нет индивидов с $N_i \gg 1$.
2. Нет стагнации, так как и к концу работы алгоритма $N_1 \neq N_2 \neq \dots$.
3. Нет необходимости в *явном вычислении пригодности*, т.к. для упорядочения индивидов достаточно иметь возможность их попарного сравнения.

- Недостатки: значительные накладные расходы на переупорядочивание и трудность теоретического анализа сходимости.

Замечание: ранговая селекция имеет мало общего с биологической.

[\[редактировать\]](#)

Турнирная селекция

- Для отбора индивида создается группа из $N(N \geq 2)$ индивидов, выбранных случайным образом
- Индивид с наибольшей пригодностью в группе отбирается, остальные - отбрасываются (турнир)
- Турнирную селекцию можно рассматривать как ранговую селекцию при наличии шума.
- Преимущества:
 1. Нет преждевременной сходимости
 2. Нет стагнации
 3. Не требуется глобальное переупорядочивание
 4. Не требуется явное вычисление функции пригодности

Замечание: турнирная селекция имеет глубокие аналоги в биологии.

[\[редактировать\]](#)

Элитарная селекция

• Как минимум одна копия лучшего индивида популяции всегда проходит в следующее поколение • Преимущество: гарантия сходимости, т.е. если глобальный максимум будет обнаружен, то ГА сойдется к этому максимуму • Недостаток: большой риск захвата локальным минимумом • Альтернатива: сохранять лучшую найденную структуру в специальной ячейке памяти и в конце эксперимента использовать ее в качестве решения вместо лучшего в последнем поколении.

[\[редактировать\]](#)

Скращивание в ГА

- *Одноточечное скращивание* представляет собой разрезание хромосом родителей в выбранной случайным образом общей точке и обмен правыми частями
- При *двухточечном скращивании* хромосома рассматривается как кольцо со связанными первым и последним генами. Кольцо рассекается на две части и полученные куски обмениваются.
- *Равномерное скращивание* предполагает, что каждый ген потомка выбирается случайным образом из соответствующих генов родителей.

Замечание: в этом случае родителей может быть больше двух, в том числе возможно участие всей популяции родителей в целом (*gene pool recombination*)

- Существует также скращивание, основанное на масках, накладываемых на хромосому, когда выполняется обмен генами, находящимися на фиксированных позициях.
- Скращивание применяется к индивидам популяции с постоянной вероятностью p_c , которая в общем случае выбирается из интервала [0.5,0.8]
- Клонирование применяется с вероятностью $1 - p_c$ (для того, чтобы в каждом поколении оставалось одинаковое количество индивидов).

[\[редактировать\]](#)

Мутация в ГА

- Мутация состоит из выполнения (обычно небольших) изменений в значениях одного или нескольких генов в хромосоме
- В двоичных хромосомах мутация состоит в инвертировании случайным образом выбранного бита генотипа, например 1010101010 \rightarrow 1011101010.
- В генетических алгоритмах мутация рассматривается как метод восстановления потерянного генетического материала (а не как поиск лучшего решения)
- В генетических алгоритмах мутация применяется к генам с очень низкой вероятностью $p_m \in [0.001, 0.01]$. Хорошим эмпирическим правилом считается выбор вероятности мутации из соотношения $p_m = \frac{1}{M}$, где M - число бит в хромосоме.

[\[редактировать\]](#)

2. Практическая часть (описание программы и порядка работы с ней)

Целью практической части моей работы стало создание компьютерной программы простого генетического алгоритма (ГА) и тестирования созданного генетического алгоритма на основе операторов.

При разработке программы я использовал следующий простой ГА:

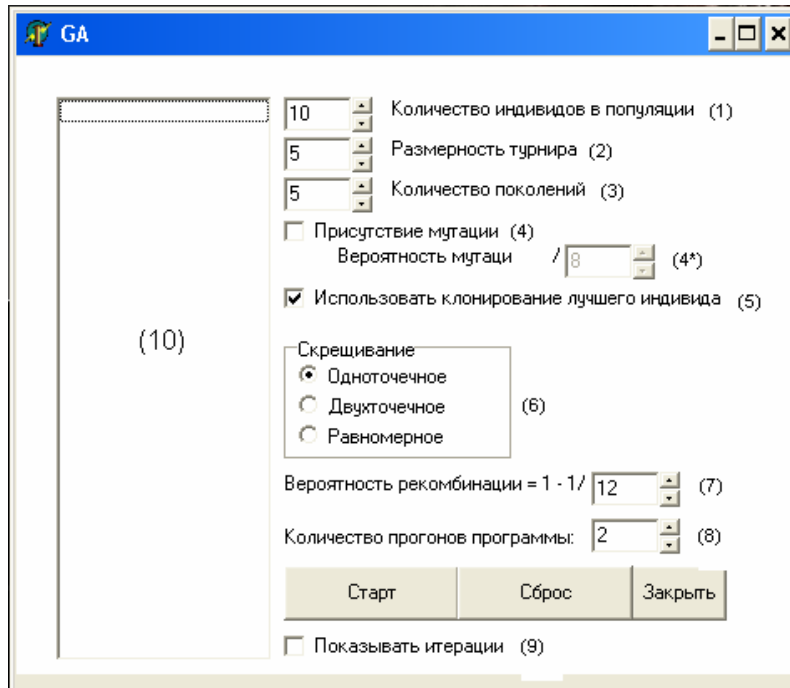
1. Генерация популяции хромосом случайным образом. Хромосома имеет 8-ми битную длину, где один бит представим в виде бинарной строки – 0 или 1.
2. Опускается процедура декодирования и, в дальнейшем, мы работаем с бинарной строкой.
3. Оценка пригодности каждого индивида происходит путём перехода с 0 на 1 (чем больше переходов, тем выше пригодность).

4. Генерация новой популяции происходит путем клонирования (копирования), турнирной селекции с выбором размерности турнира, рекомбинации и частичной мутации хромосом (значения мутации и рекомбинации задаются самостоятельно).
5. Условие останова опускается (т.к. стоит конкретная задача усреднить результаты работы ГА и на основе их сделать таблицу).

[\[редактировать\]](#)

Порядок работы с программой

Общий вид программы при запуске:



Перед запуском программы нужно установить параметры с которыми будет работать запрограммированный алгоритм.

- (1) Выбирается количество индивидов в популяции (по умолчанию 10 индивидов), т.е. то количество бинарных строк, которые изначально будут генерироваться случайным образом при каждой программной прогонке (8).
- (2) Представленный в программе ГА был запрограммирован для работы с турнирной селекцией, а пользователю остаётся выбор размерности турнира (2) (по умолчанию 5 индивидов).
- (3) Выбирается количество поколений (по умолчанию 5 поколений).
- (4) Устанавливается флажок для присутствия в ГА мутации (по умолчанию флажок не установлен, т.е. мутации при запуске не будет).
- (4*) При включенном флажке мутации(4) устанавливаем вероятность этой мутации (по умолчанию мутация равна 1/8). Условно будем считать, что средняя мутация равна 1/8, высокая равна 1/4 и низкая 1/16.
- (5) Устанавливается флажок для клонирования лучшего индивида (установлен по умолчанию). Клонирование позволяет переносить без изменения индивида с наилучшей пригодностью в следующее поколение.
- (6) Выбираем оператор скрещивания (по умолчанию одноточечное).
- (7) Выбираем вероятность рекомбинации (по умолчанию 1-1/12). Вероятность того, что выбранные индивиды будут скрещиваться.
- (8) Выбираем количество прогонов программы (по умолчанию стоит 2 прогона). Итак, мы установили основные параметры для работы алгоритма. Но перед запуском программы, для которого служит кнопка "Старт" следует определиться:
- (9) Хотим ли мы видеть итерации программы или сразу переходить к отчёту (по умолчанию флажок не стоит, а значит, мы сразу переходим к отчёту после нажатии кнопки "Отчёт"). Кнопка появляется после запуска программы.

Если же мы поставили флажок (9) для просмотра итераций, то после запуска программы мы можем просматривать итерации работы алгоритма, нажимая на кнопку "Далее".

GA

#1 10001111 (1)
#2 01001100 (2)
#3 10000110 (1)
#4 00100101 (3)
#5 00001010 (2)
#6 01010010 (3)
#7 01111111 (1)
#8 01101011 (3)
#9 00111011 (2)
#10 01100101 (3)

лучший: #10 01100101

Количество индивидов в популяции: 10
Размерность турнира: 5
Количество поколений: 5
☐ Присутствие мутации
Вероятность мутации = 1 / 8
☒ Использовать клонирование лучшего индивида

Скрещивание:
☒ Одноточечное
☐ Двухточечное
☐ Равномерное

Вероятность рекомбинации = 1 - 1 / 12
Количество прогонов программы: 5

Старт Сброс Закреть

☒ Показывать итерации

Прогон №: 1
Поколение №: 1

Далее Отчёт

Турнир# 1
01101011 (3)
00111011 (2)
10001111 (1)
00100101 (3)
01111111 (1)
победитель: 00100101 (3)

Турнир# 2
00111011 (2)
00001010 (2)
01101011 (3)
01010010 (3)
01111111 (1)
победитель: 01010010 (3)

Турнир# 3
00100101 (3)
10001111 (1)
10000110 (1)
01101011 (3)

Пара №1 (7)
00100101 00100100
01010010 01010011 01010011 (3)

Пара №2 (3)
01101011 0110010
01010010 01001011 01001011 (3)

Пара №3 (3)
01101011 01101011 01101011 (3)
01101011 01101011

Пара №4 (3)
00111011 00110010 00110010 (2)
01010010 01011011

Пара №5 (3)
01101011 01100101
00100101 00101011 00101011 (3)

Пара №6 (2)
01101011 01101011 01101011 (3)

В этом случае отчет мы можем просмотреть только после просмотра всех итераций. **Люди HELP можете ли вы предоставить эту прогу??? оч нужно для сдачи курсовика.. напишите на почту пожалуйста maxorol@yandex.ru** Сорри что испортил статью, не знаю как отправлять ЛС, если они тут есть

Категории статей: [Незавершённые статьи по эвентологии](#)

[Японские кухонные ножи Samura...](#) Назови пароль 'Викзнание' и получи скидку!

[Яндекс.Директ](#) [Все объявления](#)

[Прибыльные партнёрские программы](#) Присоединяйтесь и зарабатывайте высокие выплаты за привлечённых клиентов
invest.vipcentr.ru

[Создать сайт бесплатно за 5 минут](#) Создать сайт? Легко! Быстро! Даром! Простой конструктор.
okis.ru

[Приличный заработок в интернет](#) Ищу людей для работы через интернет! Действуйте!
quick.success.ru

[Английский за полгода!](#) Секретный метод обучения сотрудников разведки. Изучайте прямо из дома!
spyenglish.ru

Sprint® Official Site

www.sprint.com/Education

Teach & Connect With Mobile Learning Solutions At Sprint®!



навигация

- [Главная Страница](#)
- [Проект Викизнание](#)
- [Новости проекта](#)
- [Последние правки](#)
- [Случайная статья](#)
- [Помощь](#)
- [Поддержите проект!](#)

поиск

wiki-инструментарий

- [Ссылки на эту статью](#)
- [Правки в связанных статьях](#)
- [Закачать файл](#)
- [Спецстраницы](#)
- [Версия для печати](#)

Эта статья была последний раз отредактирована 17:44, 17 <декабря> 2011 пользователем [maxorol@yandex.ru](#). Используются материалы, предоставленные Викизнание - пользователи [Evgen2](#) и [Helgus](#) и Анонимные пользователи Викизнание.

Распространение разрешено на условиях [Викизнание:License](#). [О проекте Викизнание](#) [Отказ от ответственности](#)

Викизнание - свободная бесплатная энциклопедия, использующая wiki-технологию



Бесплатный и платный wiki-хостинг - место для ваших wiki-страниц и сайтов от создателей Викизнания , соционика-wiki .

Мы ищем спонсоров