会社説明会資料

interprism

インタープリズム株式会社

目次 Index

概念 Concept

データ Data

採用試験 Examination

会社の目的 Company Purpose

3 つの目的を持ち事業を運営しています

充実した勤務体験

労働を単に社会貢献や収入のための手段と諦めるのではなく、仲間とコミュニケーションを楽しみ、テクニカルスキルおよびビジネススキルを向上させ、共有する目的を達成させ、充実感を味わえる場ととらえたいと考えています。

収益

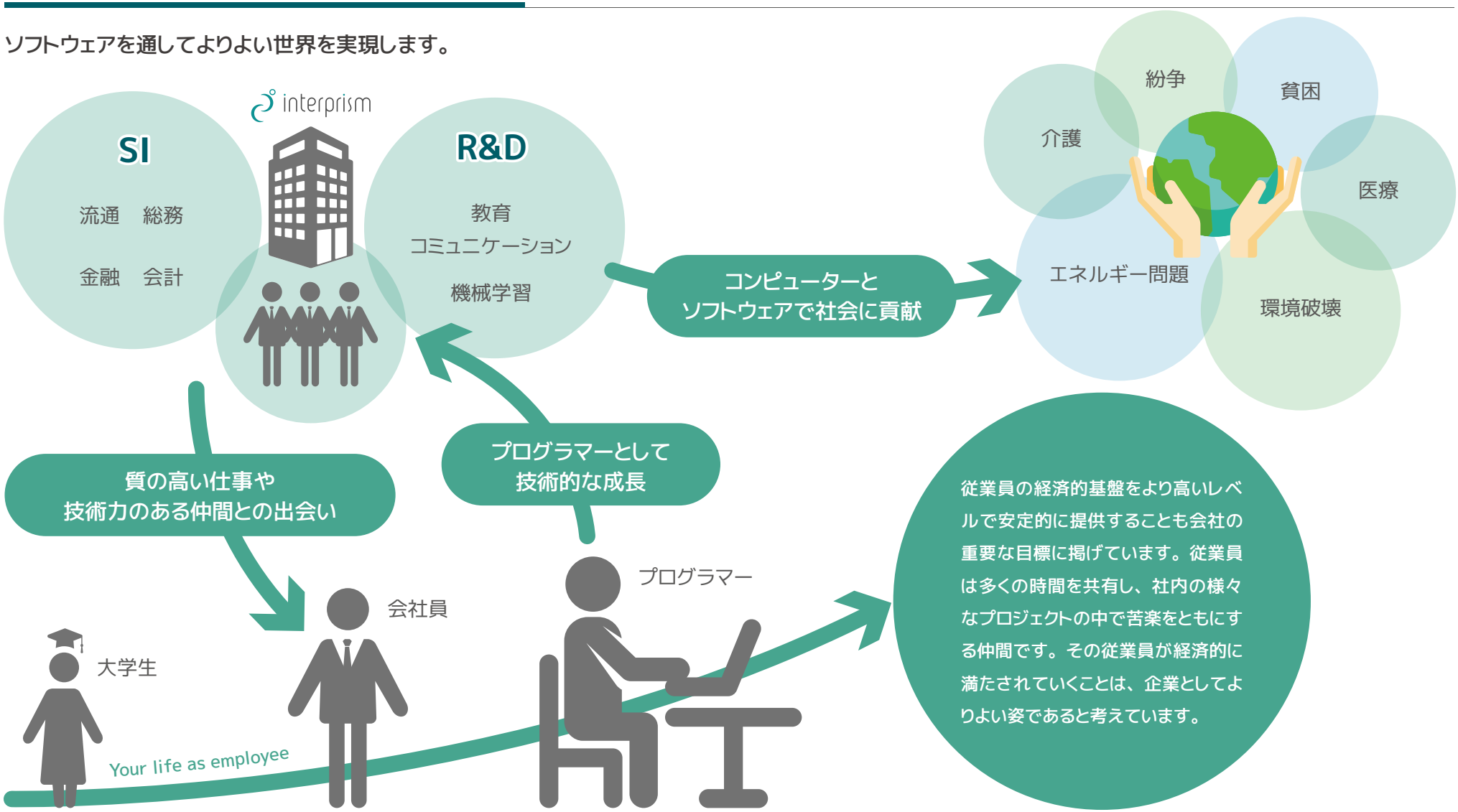
継続的に社会貢献および従業員の充実した業務体験を実現するには、安定した収益が必要になりますが、収益には業務活動の客観的評価の指標としての側面があり、この指標の最大化を目的とすることの意味および効果は小さくありません。当社は会社と従業員、双方の経済的な繁栄を目指しています。

社会貢献

当社の究極の目標は、プログラミングにより、人々を単純作業から解放して、世界中の人々がよりクリエイティブな生活を送れるようにすることです。市場経済において取引されるあらゆる商品やサービスの価格には、これらを作り出す過程で投入された労働に対する賃金が反映されています。この労働力のうち作業手順が一般化できないものはほとんどないと考えており、このことが当社がソフトウェア事業に注力する根源的理由となっています。

会社のイメージ Company Image

ソフトウェアを通してよりよい世界を実現します。



会社のビジョン Company Vision

Mission ～ 使命=存在意義

コンピューターに息を吹き込む魔術師でありたい。

プログラムを与えられたコンピュータは時に生命を宿した人のように振る舞います。

当社の究極の目標は、プログラミングにより、人々を単純作業から解放して、世界中の人々がよりクリエイティブな生活を送れるようにすることです。

ルールがあり、法則があるものは原理的にはプログラム可能であり、広義の意味では単純作業といえます。しかし、人々にとって単純作業と思われることも、その背景には1000億以上の神経細胞が何年もの時間をかけて学習した頭脳が前提になっており、これをコンピュータの動作原理であるブール代数の膨大な積み上げによって人工的に置き換えていくには、高度な知的作業が必要とされます。

この人々の"単純作業"をコンピュータによって置き換える試みは発展の途上であり、"単純作業"の領域はこの分野に参画している企業の競争により広がり続けています。

この競争の優劣がエンジニアの思考力に強く依存していることが、当社がこの分野にかける最大の理由であり、我々はこの分野でより優れたソフトウェア開発力を身につけることで、この使命を達成していきたいと考えています。

Vision ～ 展望=方向性

我々が目指す未来像は、知的で合理的な世界です。

まず自社が知的で合理的な組織になり、知的で合理的な戦略の妥当性を証明しなければなりません。

この課題に成功したとき、我々のMissionが達成されるはずです。

我々は優れた知識やアイデアが全世界で共有される時代に適した手法で、より優れたソフトウェアを開発していきたいと考えています。

数学に代表されるクラッシックな学問分野で鍛えられる思考方法を土台にして、世界中で研究、開発された実装技術、開発手法を積極的に導入することで、より高度な仕事を可能にするソフトウェアをより短時間で開発することを目指します。

ソフトウェアの設計や実装は、システムを開発するための手段ではありますが、そこで展開される様々な手法や理論は学術的にも興味深いものが多く、そういった知識を共有し、互いに刺激を与えあうことのできる社内文化を目指しています。

各分野で優れた知識と能力を持ったエンジニアが集まり、議論を重ねながら、ソフトウェア開発を進めていくことは、クリエイティブな作業であり、知的好奇心を満足させるものです。当社はより高いレベルでこの知的好奇心を満たすことのできる環境作りを目指しています

従業員の経済的基盤をより高いレベルで安定的に提供することも会社の重要な目標に掲げています。従業員は多くの時間を共有し、社内の様々なプロジェクトの中で苦楽をともにする仲間です。その従業員が経済的に満たされていくことは、企業としてよりよい姿であると考えています。

更に当社は、会社が従業員の経済的基盤を支え、充実した業務時間を提供するためだけの存在では不十分だと考えています。多くの場合、会社は従業員が最も多くの時間を費やすコミュニティの1つとなります。従業員の充実した生活を総合的サポートするために、様々な福利厚生を充実させることも会社の大きな役割だと捉えています。

ソフトウェア開発業務の本質とは思考することであり、その意味において、個々の従業員の思考力の総和が当社のソフトウェア企業としての力を表わすことになります。全社員が深く思考することで、会社業務がなりたち、充実した勤務スタイルをを確立していくことを理想としています。

Strategy ～ 戦略

企業の活動は、その企業に所属する社員の意思に基づいています。この自明の事実が企業を有機的な生き物のように振舞わさせる一つの要因になっています。従って、まず採用を強化しなければいけないのは自明な戦略となります。そして、教育にも力を入れなければなりません。高い能力と知識をもった社員は難しい課題を次々にこなしていきます。

純粋な"勉強"によっても技術知識は身につきますが、実際の開発案件の中で身につける知識は、より実践的であり、また、妥協することができない環境で開発に打ち込むことで、より深い知識や経験が身につくケースが少なくありません。その意味において、収益と同等に請け負う開発案件の"質"を見極めなければいけません。技術者にとって最適な案件をアサインすることで、技術者の成長を最大化させ、ひいてはその技術力が最大収益に繋がっていくと考えています。

当社はSI(System Integration)とR&D(Research & Development)を事業の核としており、これは今後も変わることはありません。SIにおいては、鍛え上げられた技術力で、他社では実現できない難度の高い顧客要望に応えることで、顧客から信頼を勝ち取り、安定的な収益に繋げていきます。新しい技術を積極的に導入することで、顧客満足度と自社技術力の両者の向上を図ります。R&Dにおいては、世の中に未だ存在していない新たなサービスを自ら企画して、実装して、世の中にアピールしていくことになります。成功する確率はSIに比べて低い代わりに成功した時のリターンはSIをはるかに凌ぐものになります。これら2つの事業をバランスよく推進していくことで、ビジネス規模の拡大を目指します。

抽象的思考 Abstract Thinking

抽象的思考とプログラミング能力の関係について

計算機科学において、抽象化とは、一度に注目すべき概念を減らすことで、巨大化、複雑化の一途をたどるシステム（プログラム）の実装を可能にするための技法のことです。

この概念は数学における「抽象化」からのアナロジーです。例えば、数はプログラミング言語上の概念であり、数学上の概念でもありますが、数の計算概念は数学の概念に基づいているため、実装の詳細はハードウェアとソフトウェアに依存したとしても、それが制約とはなりません。

大まかに言えば、抽象化は制御抽象化とデータ抽象化に分けられます。制御抽象化は動作の抽象化であり、データ抽象化はデータ構造の抽象化です。例えば、構造化プログラミングでの制御抽象化とは、サブプログラムや定式化された制御フローの使用を意味します。データ抽象化とは、本来ビット列であるデータを意味のある方法で扱うことを意味します。オブジェクト指向プログラミングはデータとコードを同時に抽象化する試みと見ることもできます。

原理

コンピューティングでの主な抽象化は言語の抽象化です。新たな人工言語はシステムの特定の観点を表現するために開発されています。モデリング言語は計画立案を補助します。コンピュータ言語は CPU が処理する単純命令を組み合わせて抽象的に表現することを可能にすることでプログラミングをより容易にしています。言語の抽象化は現在も続いており、例えばスクリプト言語やドメイン固有言語でこの進化が著しい状況です。

プログラミング言語では、一部機能によってプログラマが新たな抽象化を生み出せるようにしています。例えば、サブルーチン、モジュール、ソフトウェアコンポーネントなどがそれにあたります。プログラミング言語自体の機能ではないが、設計技法上の抽象化としてデザインパターンやソフトウェアアーキテクチャがあります。

抽象化によっては、次々に構築される概念を完全に隠蔽することでプログラマが把握しなければならない概念の幅を制限しようとします。

制御抽象化

制御抽象化はプログラミング言語を使う主たる目的の 1 つです。コンピュータが理解する操作は極めて低レベルであり、メモリのある場所から別の場所へ何ビットかを移動させ、2 つのビット列を加算するといったことでしかありません。プログラミング言語を使うことでこれをもっと高いレベルに変換することができます。例えば、次のようなプログラム内の文（式）があるとします。

a:=(1+2)×5

人間にとっては、これは非常に単純で明らかな計算です（1 足す 2 は 3、これに 5 をかけて 15）。しかし、これを評価するには低レベルの実行ステップに落とし込まねばならないし、計算結果である 15 を変数“a”に代入するという作業も複雑です。数値は二進数表現に変換され（これも大方の予想よりも複雑な作業である）、（コンパイラやインタプリタが）計算をステップに分解して機械語の命令列に直す（機械語あるいはアセンブリ言語は一般のプログラマにとっては直観的に理解可能なものではなく、その内容は通常の加算や乗算といった人間の考える算術とは趣きが異なります）。そして、計算結果の“15”を“a”というラベルの付いた変数に格納するのですが、実際には物理メモリか仮想メモリ上のあるアドレスのメモリ位置がそれに対応しており……などなどです。

制御抽象化なしでは、プログラマは機械語レベルでレジスタやメモリアドレスを指定してプログラムを書かねばなりません。その場合 2 つの深刻な結果を招きます。第 1 に似たような機能を毎回コーディングしなおさなくてはならなくなります。第 2 にプログラマは特定のハードウェアや命令セット向けにプログラムを書くしかなくなってしまいます。

データ抽象化

データ抽象化とは、データ型の「抽象的」属性と「具体的」実装詳細の明確な分離を強制することです。抽象化された属性はデータ型（あるいはそのインタフェース）を利用するクライアントコードに対して明確化され、具体的実装は完全にプライベートな状態でかつ必要に応じて変更できる形で存在します。概念的には、そのような変更は抽象的振る舞いには変化をもたらさないので、クライアントコードには全く影響を与えません。

例えば、「参照テーブル」という抽象データ型を定義したとする。参照テーブルには「キー」とユニークに対応する「値」があり、キーを指定することで値を操作できます。このような参照テーブルの実装方法はハッシュテーブルや 2 分探索木や線形リストなどいくつかあります。クライアントコードからすれば、データ型の抽象化された属性はどの場合でも同じです。

もちろん、以上の話は最初にインタフェースを正しく詳細化することにかかっており、そうでないと実装の変更がクライアントコードに影響を及ぼしてしまいます。別の見方をすれば、インタフェースをデータ型とクライアントコードの間で合意された振る舞いの「契約」を形成すると考えることもできます。契約にない部分は予告なく変更される可能性がある、ということになります。

オブジェクト指向プログラミング言語も一般にデータ抽象化を提供すると言われています。

■参照：Wikipedia https://ja.wikipedia.org/wiki/ 抽象化 _(計算機科学)

数学とコンピュータ Mathematics & Computer

数学とコンピュータ

当社はソフトウェア開発会社でありながら、数学的思考力を大変重視しています。それはコンピュータ科学と数学が極めて密接な関係にあることに起因していると考えています。

コンピュータにおける各技術にどのような形で数学が関係しているか、その一部を列挙したいと思います。

CPU

- ブール代数

コンピュータの中心に位置する CPU は、トランジスタによる膨大なブール代数の組み合わせによって、成り立っています。普段我々が行うプログラミング作業において、意識することはありませんが、CPU の動作原理の概要を知っていることは、高度なソフトウェア設計を行うのに役に立ちます。

データベース

- 計算理論
- 解析学

データベースは巨大なデータを読み書きするためのアプリケーションです。データ構造とパフォーマンスに関するノウハウが凝縮しています。パフォーマンスの計算は、基本的には比較的簡単な数式が利用されますが、稀に高等数学の知識を使うこともあります。

ネットワーク

- グラフ理論

インターネットに代表されるように、情報工学においてネットワークモデルをデータ構造として利用することは少なくありません。ネットワーク経路探索等の機能が必要とされるようなケースでは、グラフ理論に関する知識を直接的に用いることも少なくありません。

ソフトウェア設計

- 抽象的思考法
- 演繹的思考法
- 帰納的思考法
- 代数的思考法
- 集合論
- 線形代数
- 解析学

ソフトウェア設計ならびに、プログラミング作業における思考過程と、数学の論理展開における思考過程には、多くの共通部分があります。ソフトウェアとは関数の集合体であり、代数的な思考そのものといえます。逐次実行、代入、関数呼び出し、条件分岐は演繹的な思考であり、プログラムに頻出する再帰的なコードの読み書きには帰納的な思考方法が必要になります。また、巨大なソフトウェアシステムを構築する際には、あまりにも多い場合分けを洗いざらい列挙するわけにはいかないため、まずは全体の機能を抽象的にとらえ、モジュール構成を設計し、徐々に徐々に具現化していかなければ、なりません。

また、直接的に数学の知識を利用するケースも多々あります。その代表格は集合論、線形代数、解析学ではないかと思います。

歴史的背景

情報科学の父とも呼ばれる論理回路を考案したクロード・シャノン。

チューリングマシンと呼ばれる仮想マシンを考案し、コンピュータの基本原理を考案したアラン・チューリング。

ノイマン型コンピュータと呼ばれる現代のコンピュータ・アーキテクチャをの原型を考案したフォン・ノイマン。

現代の情報社会に極めて大きな役割を果たした上記3人の科学者はいずれも優れた数学者であることからも、情報工学の背景には数学による裏付けがあることが感じられます。

ビジネスモデル Business Model

System Integration

広範囲なIT知識を生かしたシステム構築業務

1 要件定義 Requirement Definition

顧客から要件を聞き出す。

2 設計 Design

要件を満たすために外部と内部の設計をする。

3 実装 Implementation

設計書に従って画面や処理を作り上げる。

4 メンテナンス / 運用 Maintenance/Operation

運用の手伝い、導入のサポートを行う。

Research and Development

SIで磨いた先進的な技術をアウトプット

WT 勤怠アプリ

Flash

Technology

実装 Implementation

インタープリズムの強みは実装力と考えています。独学ではなかなか短期間では学べないことでも、先輩社員による指導や教育ノウハウ、
そしてハイスペックなマシンなど環境を万全に整えることで、若手社員の能力を引き出します。

1 Latest Technology

最新の技術を取り入れる

インタープリズムの案件獲得の優先順位

1. 技術選定まで任せてもらえる案件
2. 新技術を扱う案件
3. レガシー技術を扱う案件

2 Development Environment

マシン等最新のスペックで開発することができる

※イメージです。

○常駐先で支給されるPCのスペックが低いことも。
○遅いビルドが業務時間を圧迫。最新PCを持ち込むことも。
○速いPCは思考を妨げない。
○下がり続けるPCの価格対性能比からも環境投資は積極的に行う。

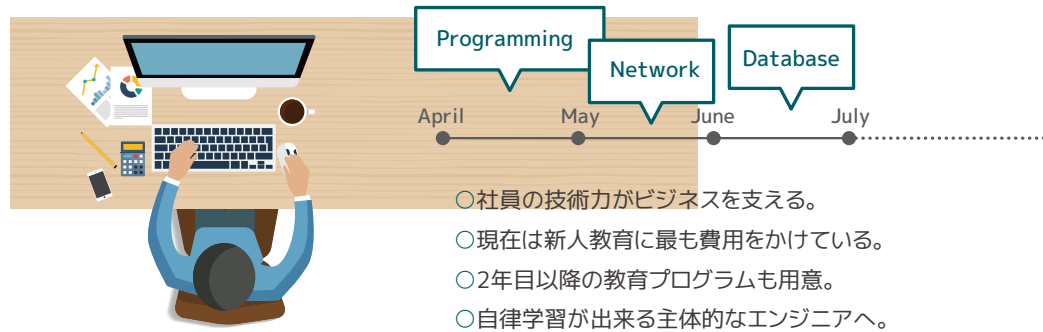
3 Review

開発したものに対し誤りや不具合、要求を満たしているかなどフィードバックし、品質を高める。

○納品するソースコードは必ず複数人でレビュー。
○断片的な知識以上のノウハウ。
○レベルの高いエンジニアとの交流が技術力向上に。

4 Training

研修期間設け、技術を身につける。



○社員の技術力がビジネスを支える。
○現在は新人教育に最も費用をかけている。
○2年目以降の教育プログラムも用意。
○自律学習が出来る主体的なエンジニアへ。

インタープリズムが求める人物像 Interprism Spirits

As a Team

チームでの達成を考える

個人が独立して動くのではなく、共有された目的のために、チームのメンバーとして有機的に機能した方が生産性が高いと考えています。

Improvement

常に自分を磨く・向上心を持つ

ITは現代の技術革新の中心的存在であり、技術革新と厳しい競争が途絶えることはありません。この事実は継続的学習の十分な動機になりえることであり、成長のチャンスととらえることができます。

Recognition as own issues

理想の状態は自分自身で掴み取る

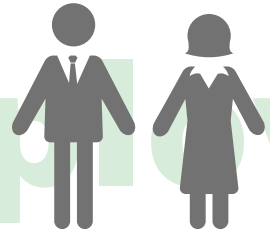
社員は会社の一部です。自分以外のだれかの働きにより理想的な状態が偶然訪れるのを待ち続けるのではなく、自分自身に何ができるかを考えることで、自身の目的の達成確率および期待値は高まります。

Respect anyone

他者を尊重する

自分および相手の立場にかかわらず、常に他者を尊重する気持ちを持つことで、無益な感情的争いを回避し、共同作業をより生産的に行えるようになると考えています。

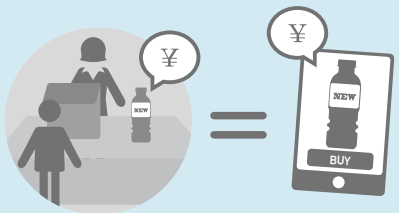
自立した精神をもつ社員一人ひとりが有機的に、粘り強く協力しあうことで技術革新が激しいIT産業の中で厳しい競争に勝ち抜くことができ、
各社員の利益を最大化させることができると信じています。



インタープリズムが求めるエンジニア像 Interprism Engineer

洗練された抽象的思考力

より上位の概念を捉えた抽象的な理解により、適切な設計や問題解決への可能性を高めます。



深い論理的思考力

「なぜそのように作るのか」- 客観的そして論理的に説明できるよう日頃からの考え抜く姿勢が求められます。

幅広い知識

ソフトウェアは、たとえ小規模なものであっても非常に多くの技術によって構成されています。

豊かな創造力

プログラミングとはものづくりであるという観点がとても重要です。

テクニカルドメイン Technical Domain

ソフトウェア開発で用いる技術は多種多様です。
より多くのプロジェクトで活躍できるようになるためには、扱うことができる領域は広げておかなければなりません。

開発言語	フレームワーク	データベース	ミドルウェア	IDE	ソースコード管理
Java / C# / PHP / Perl / Scala / JavaScript / TypeScript / Haskell / Node / Ruby / Swift / C++ / Python / Objective-C / Kotlin	JSONIC / Angular / D3.js / Laravel / Jersey / Vue / React / Weld / Mithril	Oracle / PostgreSQL / SQL Server / SQLite / LimGraph / MongoDB / Memcached / MySQL	Tomcat / GlassFish / WildFly / Browserify / Gulp / Deeplearning4j / Node.js	IntelliJ IDEA / Eclipse / Android Studio / Xcode	Git / GitLab / SVN

取引先 Customer

株式会社 IIJ テクノロジー
株式会社アイオス
株式会社アイ・ティー・ワン
株式会社 iTiD コンサルティング
アットホームホールディングス株式会社
アルタイル株式会社
アローシステムズ株式会社
株式会社 NTT データ関西
株式会社 NTT データクリエーション
株式会社エフネット
株式会社O2
ソニー株式会社
ソニーイメージングプロダクツ&ソリューションズ株式会社
ソニーグローバル M&O 株式会社
ソニーセミコンダクタソリューションズ株式会社
ソニーマーケティング株式会社
東芝デジタルソリューションズ株式会社
株式会社DCS
株式会社電通国際情報サービス
ビッグローブ株式会社
フューチャーアーキテクト株式会社
丸紅情報システムズ株式会社
株式会社 LIGHTz

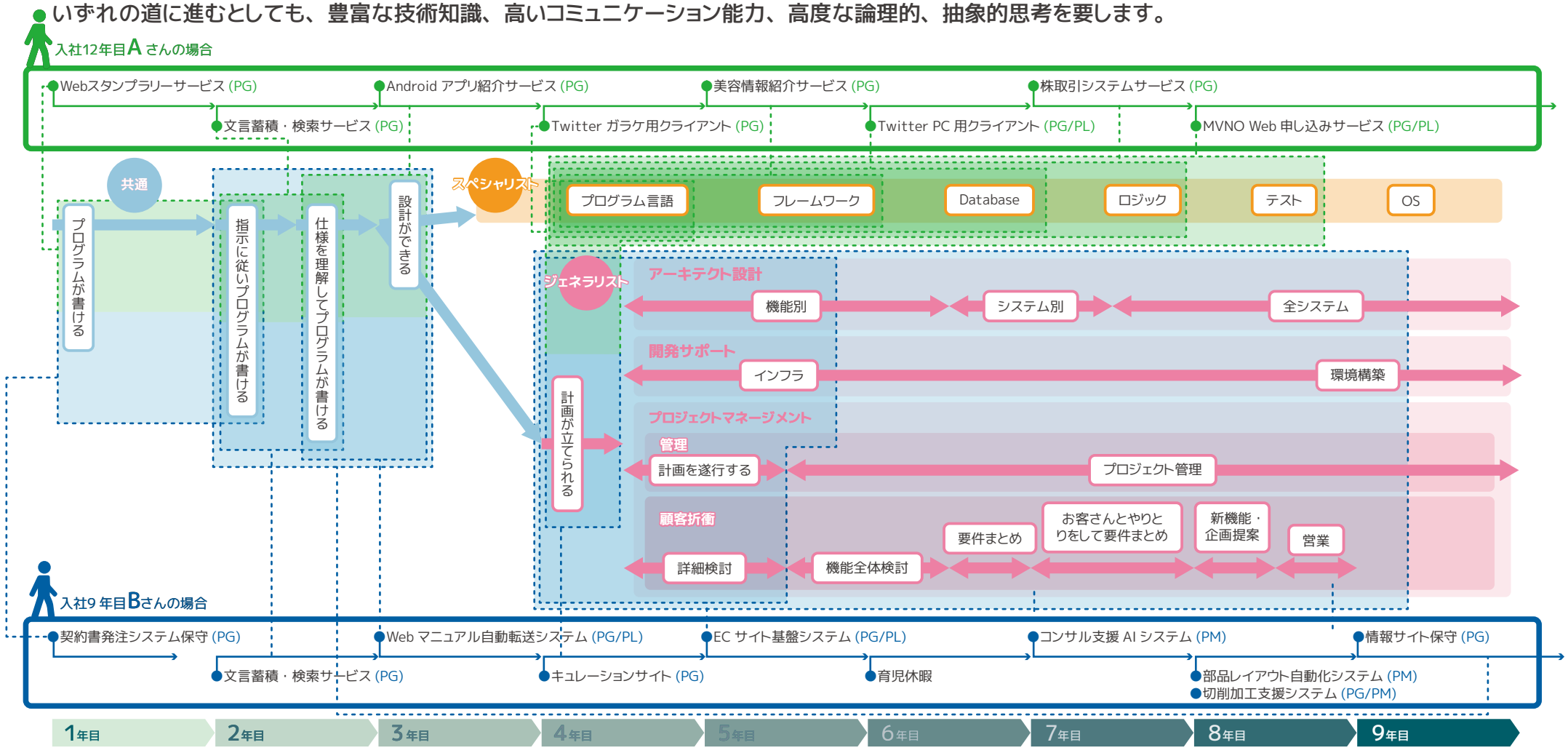
など ※あいうえお順

※2021年現在の主な出向先です。
案件により変更になる可能性があります。

キャリアパス Career Path

3年目以降は様々なキャリアパスが考えられます。

いずれの道に進むとしても、豊富な技術知識、高いコミュニケーション能力、高度な論理的、抽象的思考を要します。



案件紹介※ほんの一部 Projects Introduction

I社

製造業者向け / ツール開発 / データベース

コンサルティング支援ツール開発

大規模製造業向けのコンサルティング支援ツールを開発します。10年以上続く大規模プロジェクトであり、数社のSI企業が参画していますが、その中でも当社が最も長く貢献しているプロジェクトです。業務内容は複雑で、技術レベルは極めて高く、全貌を理解するには数年が必要となります。プログラミング言語としてはC#、Javaを主に使います。また、極めて高度なデータベースの知識が必要になります。

言語	C#, Java, PowerShell, SQL
DB	SQL Server
開発ソフト	Microsoft Visual Studio SQL Server Management Studio (SSMS)、Eclipse Visual Studio Code
その他	.NET Framework, Subversion Redmine, Slack, iQUAVS Git併用も模索中
クラウド	Azure

L社

AIシステム開発 / ベンチャー企業 / 技術サポート

AI思考パターン可視化分析システム開発

AIシステム開発を手がけるベンチャー企業のソフトウェア実装を技術的にサポートします。技術的には主にJavaScript、Java、を扱います。大規模システム開発と違い、抽象度の高い要件に対して、即座に要求分析をして、実装実現性を調査し、実装して動くものを作っていかなければなりません。エンジニアとしての総合的な力が試されます。

言語	Java, TypeScript, WebGL
フレームワーク	Mithrill.js, PixiJS, Jetty
DB	LimGraph
バージョン管理	Git

B社

製造業 / 支援ツール / 技術サポート

部品レイアウト自動化システム開発

製造業における作業支援ツールを開発しています。Pythonを使ったプロジェクトで、画像解析で特徴の抽出を行ったり計算によって部品の組合せを決め、関連するシステムとの連携を行い蓄積データから最適な解を導きだします。
研究開発として取組む部分も多いのでトライアンドエラーで確認しながら素早いアウトプットを求められます。

言語	Python
その他	画像解析(Open-CV)
	BLF法

S社

半導体メーカー / システム開発 / 保守運用 / インフラ

製品開発課題分析システム開発

当社が単独で担当する小規模SIプロジェクトで、業務内容と触れる技術が多様です。開発では、顧客との要件定義、設計実装、導入保守運用...SI開発プロセスほぼ全てに参画し、ミドルウェアの導入管理やOS環境移行等、インフラに触れる機会もあります。最新技術は追いませんが、基盤から広く深く身に付く環境です。

言語	Ruby, RoR4
DB	MySQL, PostgreSQL
フロント	JavaScript, css
ミドルウェア	Apache, Phusion Passenger
開発環境	Docker

LIMGRAPH

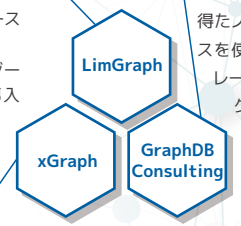
グラフ構造でデータを保存する新しいタイプのデータベース

ミドルウェア

グラフ構造を高速に永続化することを可能にする、データベースエンジンです。
LimGraph社のコアテクノロジーとして、様々なシステムに導入されます。

サービス

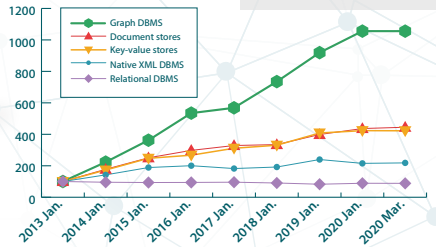
グラフデータベース開発経験から得たノウハウをグラフデータベースを使ったシステム・インテグレーションの現場にフィードバックします。



ツール

ハイパフォーマンス、イージーモデリングといったグラフデータベースの特徴を生かして、企業活動を活性化させる様々なシステムとツールをご提供していきます。

LimGraph	グラフデータベース
Flex	エクセルデータ連携システム
SemanticGraph	ファイル自動整理システム
BOMGraph	製造業向けBOM管理システム
Structure	製造業向け手順書作成システム



社名
LimGraph株式会社

創業・設立
2018年（平成30年）4月2日

資本金
3,200万円

代表取締役社長
青木 規彰

事業内容
・グラフデータベース及び関連ソフトウェアの販売
・グラフデータベースのシステムコンサルティング

プロジェクト例 Project Example

案件概要(業務仕様)

販売管理システム

✓インターネットで商品を販売する企業の販売管理システムを構築・機能改善します。

技術

✓言語／ Java, Groovy
✓DB ／ Oracle, MySQL
✓ライブラリ、フレームワーク／ Spring Framework, Spring Boot, Spock（テスティングフレームワーク）, MyBatis（ORM）, Javaslang, Vavr
✓その他ミドルウェア／ Tomcat

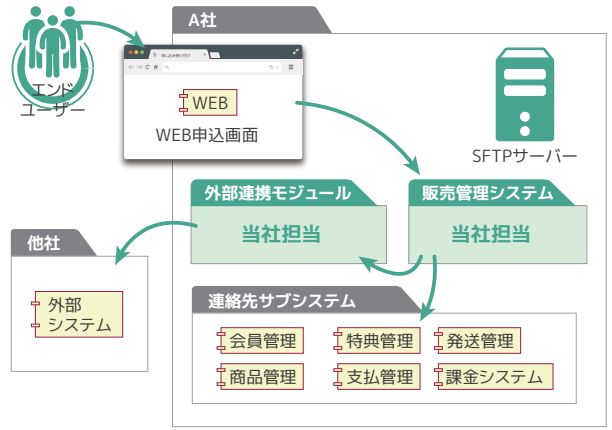
要求される技術(能力)

✓最低限Javaが書ける
✓連携先のIF仕様書等を読んで、正確に仕様を把握し、問題点を見つけることができる。
✓打ち合わせ内で色々決まるのでコミュニケーション能力が必要。
✓改善提案をどんどんできる

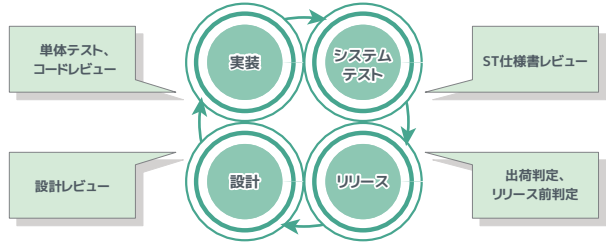
この案件で獲得できる技術・知識

✓DDD（ドメイン駆動設計）
✓コミュニケーション能力
✓大規模開発のノウハウ

システムアーキテクチャ図

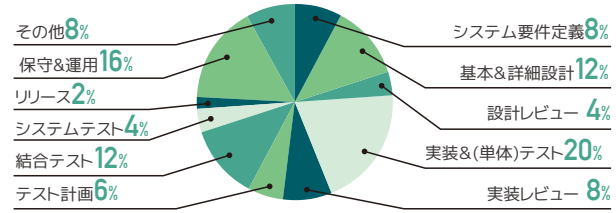


開発プロセス



グループ体制

勤務時間の内訳



経歴別タスク

	PM 7年目	SE 5年目	SE 5年目	SE 2年目
見積・請求(事務)	○			
工数見積	○	○	○	○
スコープ調整	○	○		
体制検討	○			
進捗管理	○			
システム要件定義	○	○	○	
基本&詳細設計	○	○	○	○
仕様調整(対顧客)	○	○	○	
新規参画者サポート			○	
仕様確認(社内)	○	○	○	○
プログラミング	○	○	○	○
コードレビュー	○	○	○	○
テスト	○	○	○	○
運用	○	○		

その他タスク

✓**環境整備**
- 開発PC環境
- ミドルウェア環境
- テスト環境
- 構成管理
- ビルド環境

✓**チーム力向上のためのアクション**
- 勉強会企画
- 懇親会企画

チーム活動

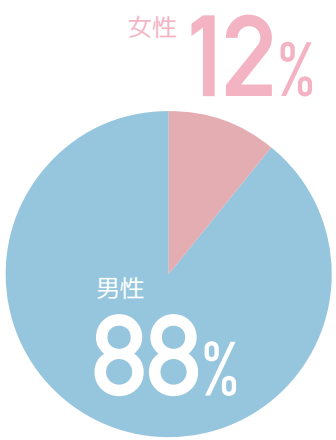
振り返り・プランニング	2時間×1回/週
デイリーミーティング	30分×1回/日
勉強会	1時間×2回/週

過去4年の社員の実績 Achievement of Employees of Last 4 years

No	入社年	現在の業務	学歴			入社前のプログラミング経験
1	2019	データー元管理システム構築支援	**九州大学大学院**	数理学府	数理学専攻	
2	2019	携帯会社ポータルサイト開発	博士 **芝浦工業大学大学院**	理工学研究科	機能制御システム専攻	
3	2019	贈り物ランキングサイト開発	**千葉大学大学院**	教育学研究科	学校教育学専攻	
4	2019	携帯会社ポータルサイト開発	**東京工業大学大学院理学院**	数学系	数学コース修士課程	
5	2019	モバイルサービス開発支援	**東京大学大学院**	工学研究科	精密工学専攻	
6	2019	贈り物ランキングサイト開発	**立教大学**	理学部	数学科	
7	2020	携帯会社ポータルサイト開発	**金沢大学大学院**	自然科学研究科	数物科学専攻	
8	2020	グラフデータベース製品開発	**京都大学大学院**	理学研究科	物理学・宇宙物理学専攻	
9	2020	携帯会社ポータルサイト開発	**慶応義塾大学大学院**	理工学研究科	基礎理工学専攻	
10	2020	モバイルサービス開発支援	**神戸大学大学院**	理学研究科	数学専攻	
11	2020	モバイルサービス開発支援	**島根大学大学院**	総合理工学研究科	総合理工学専攻	
12	2020	モバイルサービス開発支援	**筑波大学大学院**	数理物質科学研究科	数学専攻	
13	2020	AI 思考パターン可視化分析システム開発	**筑波大学大学院**	数理物質科学研究科	数学専攻	
14	2020	AI 思考パターン可視化分析システム開発	**東京工業大学大学院**	工学院	経営工学系	
15	2020	贈り物ランキングサイト開発	**東京大学大学院**	理学系研究科	物理学専攻	
16	2020	携帯会社ポータルサイト開発	**東北大学大学院**	理学研究科	数学専攻	
17	2020	AI 思考パターン可視化分析システム開発	**広島大学大学院**	理学研究科	数学専攻	
18	2021	データー元管理システム構築支援	**秋田大学大学院**	理工学研究科	数理・電気電子情報学専攻	
19	2021	携帯会社ポータルサイト開発	**京都大学大学院**	理学研究科	数学・数理解析専攻	
20	2021	モバイルサービス開発支援	**京都大学大学院**	理学研究科	物理学・宇宙物理学専攻	
21	2021	モバイルサービス開発支援	**東京大学大学院**	理学系研究科	物理学専攻	
22	2021	AI 思考パターン可視化分析システム開発	**東京都立大学大学院**	理学研究科	数理科学専攻	
23	2021	携帯会社ポータルサイト開発	**東北大学大学院**	理学研究科		○
24	2021	AI 思考パターン可視化分析システム開発	**広島大学**	理学部	数学科	
25	2021	AI 思考パターン可視化分析システム開発	博士 **広島大学大学院**	理学研究科	数学専攻	
26	2021	贈り物ランキングサイト開発	**横浜国立大学大学院**	理工学府 数物・電子情報系理工学専攻		
27	2022	新人研修（4 ヶ月予定）	博士 **大阪市立大学大学院**	理学研究科	数物系専攻	
28	2022	新人研修（4 ヶ月予定）	**埼玉大学大学院**	理工学研究科	数理電子情報系専攻	
29	2022	新人研修（4 ヶ月予定）	**東京大学大学院**	数理科学研究科	数理科学専攻	
30	2022	新人研修（4 ヶ月予定）	**東京都立大学大学院**	理学研究科	数理科学専攻	
31	2022	新人研修（4 ヶ月予定）	**名古屋大学大学院**	情報学研究科		○

データでみるインタープリズム Data Reference

男女比



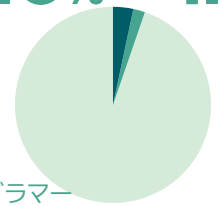
平均年齢

32歳

社員構成

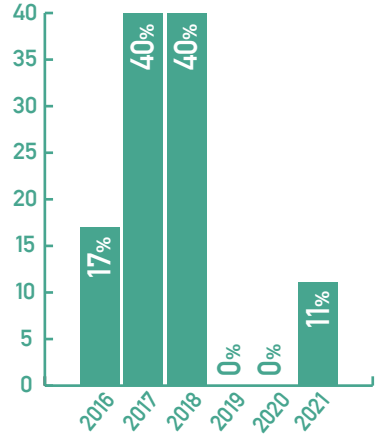
総務・事務
3.6%

デザイナー
1.8%



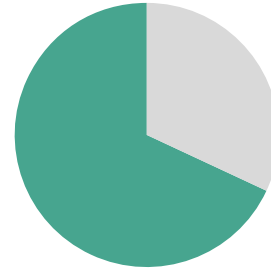
プログラマー
94.6%

プログラミング経験入社

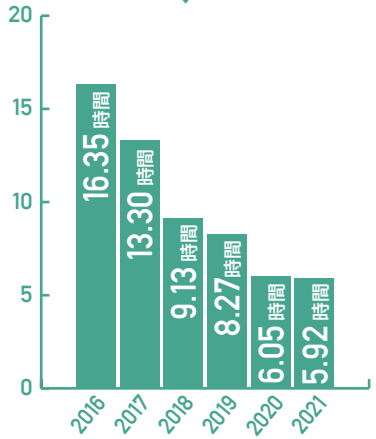


有休取得率

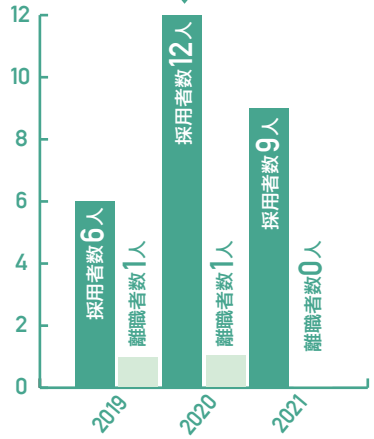
消化率
68%



月平均残業時間

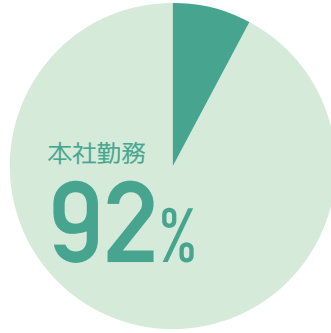


離職率



勤務地

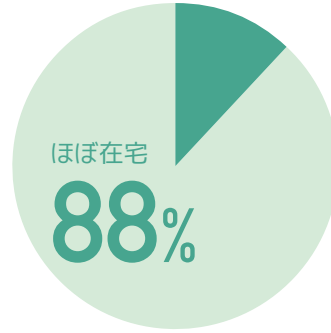
客先勤務
8%



本社勤務
92%

在宅ワーク率（変動あり）

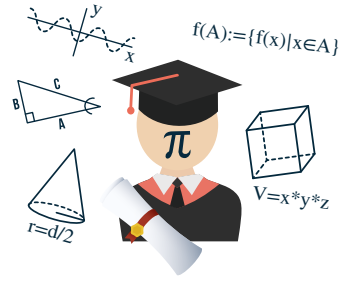
ほぼ出社
12%



ほぼ在宅
88%

数学系学部卒率

42%



Face to Face Quarter Interview

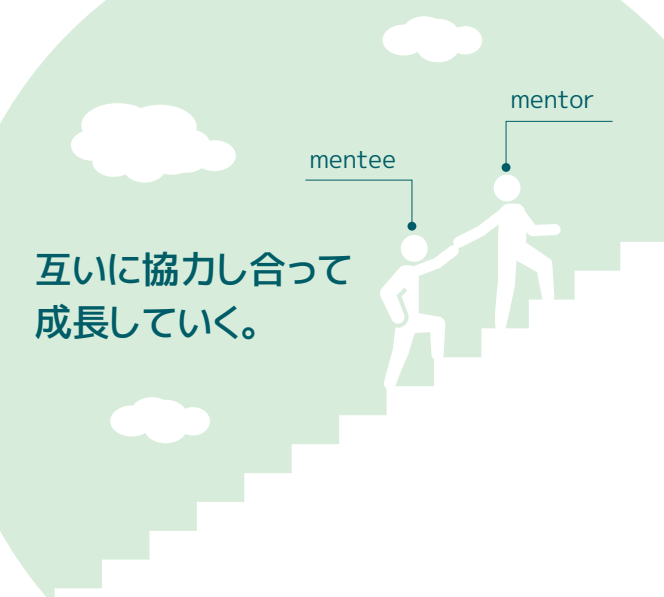
コーチングの本質的な効用やそのメカニズムが完全な形で解明されているわけではないと思うのですが、少なくともコーチングが単にコーチが持っている知識を授けるだけのものでないということ、そして、コーチングには何らかな肯定的な効用があるということはわかっています。
当社は組織的にこの手法を採用していくことで、個人ならびに会社のパフォーマンスを最大化させていきたいと考えています。

01 入社～グループ配属（約9ヶ月）

入社直後は新しい組織の文化に触れ、期待とともに、最も不安およびストレスが大きくなる時期と言えます。入社直後から各新入社員にはマンツーマンでアドバイザーがつきます。アドバイザーは自身が参画する開発プロジェクトをこなしながら、新入社員とメッセージアプリでコミュニケーションを図り、少なくとも月に1回はFace to Faceで面談を行うことを義務付けられています。新人社員は、技術的なことから新人研修のこと、研修後のプロジェクトのことから社内の些細な制度のことまで気兼ねなくアドバイザーに質問することができます。

02 グループ配属以降

グループ配属後は、グループマネージャがアドバイザーの役割を担当します。グループマネージャはグループメンバーの成長が最大になるように尽力しなければなりません。言うまでもなく成長のために実際にアクションを実行するのは、本人意外にありません。グループマネージャは4半期に一度、メンバーと面談を実施することを義務付けられており、その中で本人から課題を聞き出したり、目標設定を行い、グループメンバーの成長をサポートするための良き”理解者”になることを求められます。
コーチとプレーヤーは決して上下の関係ではありません。適切に役割分担をすることで、プレーヤーの最大パフォーマンスを引き出すための1つの”体制”にすぎません。特にコーチはこのことを理解して、自己の主張をプレーヤーに押し付けるのではなく、プレーヤーから最大パフォーマンスを引き出すことに専念しなければなりません。



年間スケジュール（新入社員） Year-round Schedule

※研修・案件期間には個人・年度により差があります。

研修内容 Training Contents

研修項目	2021年新人研修内容	2022年新人研修内容	2023年新人研修内容
技術研修	A）Java基本文法 B）基本アルゴリズム C）Collections APIとデータ構造 D）Java IO API E）テスト基礎 F）プログラミングコンテスト G）Web技術基礎(HTML,CSS) H）JavaScript基礎 I）データベース基礎 J）UNIXおよびインフラ技術基礎 K）Webアプリ開発(仮想案件)	A）Java 基本文法 B）基本アルゴリズム C）Collections API とデータ構造 D）Java IO API E）テスト基礎 F）プログラミングコンテスト G）Web 技術基礎（HTML,CSS） H）JavaScript 基礎 I）データベース基礎 J）UNIX およびインフラ技術基礎	A）Java 基本文法 B）基本アルゴリズム C）Collections API とデータ構造 D）Java IO API E）テスト基礎 F）プログラミングコンテスト G）Web 技術基礎（HTML,CSS） H）JavaScript 基礎 I）データベース基礎 J）UNIX およびインフラ技術基礎
合宿	A）設営 B）開式の辞 C）Ice Break D）7 つの習慣 E）限定コミュニケーション F）リーダブルコード輪読 G）見積もり方法 H）キャリアケーススタディ I）課題分析	A）設営 B）開式の辞 C）Ice Break D）7 つの習慣 E）限定コミュニケーション F）リーダブルコード輪読 G）見積もり方法 H）キャリアケーススタディ I）課題分析	A）設営 B）開式の辞 C）Ice Break D）7 つの習慣輪読 E）見積もり方法 F）限定コミュニケーション G）キャリアケーススタディ H）課題分析 I）レビュー練習

雇用条件 Hiring condition

休日・休暇

<休日> 完全週休2日制、祝祭日、夏季休暇（※7月～ 9月の間で3日間）、年末年始、年間休：約125日（年ごとに変動あり）

<有給休暇> 入社初年度は入社月に応じて最大10日、入社次年度以降～最大20日

取得率：約68%（2021年度）

<特別休暇> 産前産後休暇、育児休暇、就業時間短縮制度（育児）、介護休暇、就業時間短縮制度（介護）

復職支援制度あり（就業時間短縮勤務等）、結婚特別休暇、慶弔休暇　等

福利厚生

- 健康診断（年1回）
- 全社懇親会（年2回）
- その他懇親会（随時）
- 花見、BBQ大会、スポーツ大会
- 給与先払い制度
- 社内活動支援（申請制度あり）
- Android端末支給
- インターネット常時接続通信費（上限5千円／月・条件付）
- 交通費支給（上限4万円／月）
- ディズニーリゾートチケット割引購入あり
- リゾート施設・スポーツ施設利用補助

  トスラブ（リゾートハウス）箱根、館山、湯沢、コナミスポーツクラブ、セントラルスポーツ、へるすぴあ、その他全国に契約保養施設あり
- 資格取得補助制度、書籍購入費＆研修／セミナー参加費支援制度

社会保険制度

健康保険、厚生年金保険、雇用保険、労災保険

給与

[入社～研修]
月給20万円（学部卒）
月給22万円（修士卒）
月給23万円（博士卒）
※研修中はみなし残業無

[研修修了後]
月給27万円～31万円（学部卒）
月給29万円～33万円（修士卒）
月給30万円～34万円（博士卒）
※みなし残業10時間／月含む

[研修期間]
2017年　約5ヶ月
2018年　約6ヶ月
2019年　約5ヶ月
2020年　4ヶ月
2021年　4ヶ月

※研修修了後給与については、左記の下限（学部卒27万円、修士卒29万円、博士卒30万円）とし、特に優秀と判断した場合に下限以上の昇給を適用するものとします。
※賞与：支給あり
※昇給：年1回（1月）　※ただし、初年度の1月の昇給なし
※みなし残業の10時間を超える場合には、別途残業手当が支給されます。
※原則研修中は残業はありませんが、残業があった場合には別途残業手当が支給されます。

勤務時間

時差出勤制。実働時間8時間／ 1日（休憩1時間）
コアタイム11：00 ～ 18：00（客先勤務の場合は客先の勤務形態に準じる）
残業あり：みなし残業10時間／月（研修終了後）、2021年平均残業時間：5.92時間／月

※研修中の勤務時間は10：00 ～ 19：00実働時間8時間／ 1日（休憩1時間）となります。
※現在はプロジェクトごとに在宅が可能である場合、個人の判断で在宅で勤務することが可能になっています。

勤務地

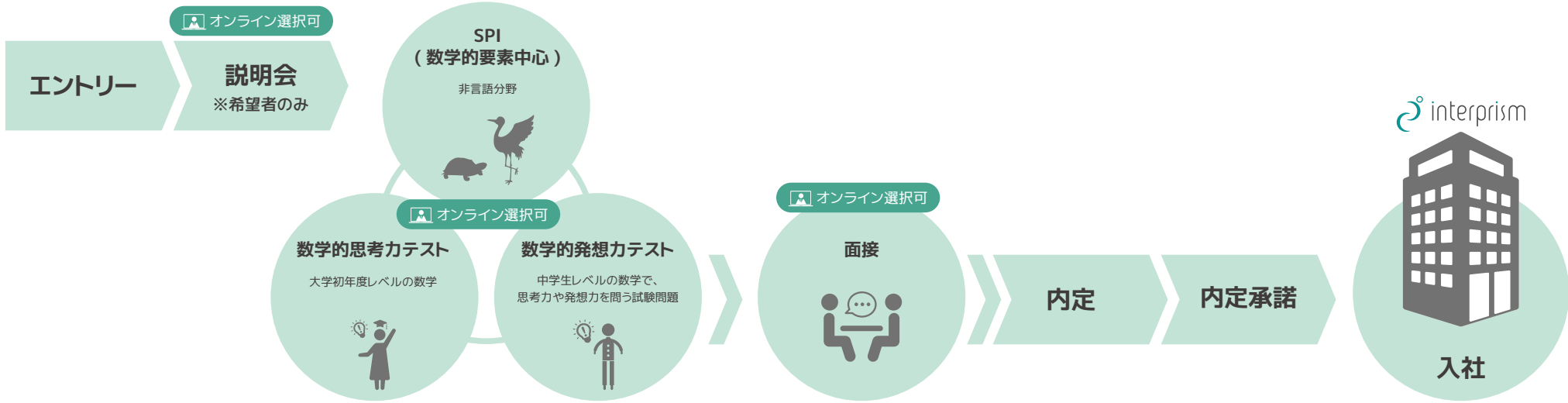
[本社勤務]
東京都中央区築地2-1-11　ガリシア銀座イースト

[客先勤務]
東京23区エリア

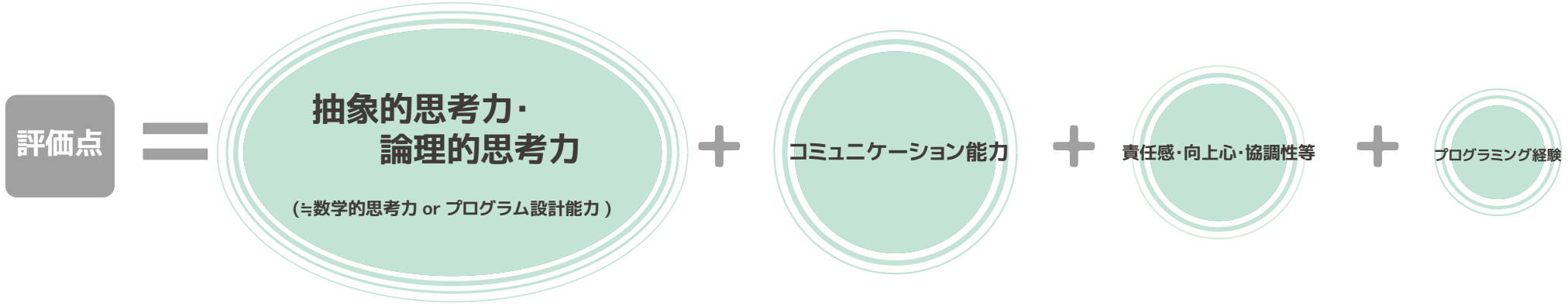
[在宅勤務]
プロジェクトによって在宅勤務が可能な場合
在宅での業務が可能な環境である場合
外部モニター貸与あり・在宅手当あり

※場所についてはセキュリティの観点から制約があります。
※月に数回の急な出勤に対応できることが条件
※入社１年目は緊急事態宣言など特別なことがない限り本社勤務

採用プロセス Employment Process



採用ポリシー Employment Policy



interprism

www.interprism.co.jp

〒104-0045
東京都中央区築地 2-1-11　ガリシア銀座イースト

TEL　03-6264-1661
FAX　03-6264-1662